ANALOG IMPLEMENTATIONS OF LOCALLY RECURRENT ADAPTIVE FILTERS

By

JUI-KUO JUAN

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1998

Copyright 1998

by

Jui-Kuo Juan

This is a dedication to my wife, Wen-Ching, and son, Andrew.

ACKNOWLEDGEMENTS

The author express his great appreciation to his committee chairman, Dr. John G. Harris, and cochairman, Dr. Jose Principe, for their continued guidance, support, and help in developing this research and giving much needed insight into the difficulties I encountered. Appreciation is also extended to Dr. Robert Fox, Dr. LiMin Fu, and Dr. Mark Yang for their interest and participation on my supervisory committee. I also would like to thank my colleagues in the CNEL Lab for their friendship and their sharing of knowledge. Finally, I extends my appreciation to my family, who have patiently endured while providing strength and encouragement to complete this work. Their love was greatly needed and will always be remembered.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

LIST OF FIGURES vii

ABSTRACT xii

CHAPTERS

1 INTRODUCTION 1

1.1 Motivation for Analog Implementation of Adaptive Filters 1
1.2 Discrete-time Adaptive Filters 2
1.3 Continuous-time Generalized Feedforward Filters 3
1.4 Sub-threshold Region 5
1.5 Previous Work 6
1.5.1 State-space Adaptive Recursive Filters 7
1.5.2 Analog Implementation of Neural Networks 8
1.6 Organization of this Work 9

2 CONTINUOUS-TIME ADAPTIVE FILTER 12

2.1 Wiener-Hopf Solution 13
2.2 The Learning Rate and Convergence Time Constant 16
2.3 The LMS Algorithm 18

3 CONTINUOUS-TIME GAMMA FILTER 22

3.1 The Gamma Kernel and Gamma Filter 22
3.2 The LMS Algorithm 27
3.3 DC Offset Effects in the LMS Algorithm 29
3.4 The Leaky LMS Algorithm 30
3.5 Circuit Implementations 31
3.5.1 Gamma Transfer Function 31
3.5.2 Wide-range Gilbert Multiplier 33
3.5.3 Adaptive Weight Circuit 35
3.5.4 Absolute-value Circuit 38
3.5.5 Current Mirror 39
3.6 Chips Test Results 41
3.6.1 System Identification 43
3.6.2 Noise Cancellation 50

4 CONTINUOUS-TIME LAGUERRE FILTER 54

4.1 The Laguerre kernel and Laguerre filter 55
4.2 Circuit Implementation of the Laguerre Filter 58
4.3 Simulations and Test Results 60

5 MULTI-SCALE CONCEPT 66

5.1 Continuous-time Multi-scale Gamma Filter 67
5.1.1 Circuit Implementation 71
5.1.2 Simulations and Measurement Results 72
5.2 Multi-scale Laguerre Kernels 78
5.2.1 Multi-scale Laguerre Circuit Implementation 81

6 MULTI-SCALE DISCRETE-TIME ADAPTIVE FILTER 83

6.1 Single Time Scale Gamma/Laguerre Filters 83
6.2 Multi-scale Gamma Filter 90
6.3 Simulations 91
6.3.1 System Identification 91
6.3.2 Echo Cancellation 94

7 CONCLUSIONS AND FUTURE WORK 99

APPENDIXES

A DC Offsets in LMS Algorithm 102

REFERENCES 106

BIOGRAPHICAL SKETCH 110

LIST OF FIGURES

1.1 Discrete-time transversal filter. 2

1.2 Continuous-time generalized feedforward transversal filter. 4

2.1 An approximated continuous-time ideal delay transversal filter using bilinear transform. 14

2.2 Adaptation of w_0. (a) Top, $\rho = 1$. (b) Bottom, $\rho = 0.01$. 20

2.3 Adaptation of w_0. (a) Top, $\rho = 0.5$. (b) Bottom, $\rho = 0.01$. These results show that the misadjustment is increasing as the learning rate is increasing. 21

3.1 Continuous-time gamma kernels. 23

3.2 Continuous-time gamma memory structure. 24

3.3 (a) Modified transconductance amplifier. (b) Symbol of transamp. . 31

3.4 The linear range comparisons of transamps with/without extra diodes. solid line : without diodes, dashed line : with diodes. 33

3.5 Modified wide-range Gilbert multiplier. 34

3.6 (a) Output currents as a function of $(w - w_{ref})$ for several values of V-V_{ref}. (b) Output currents as a function of (V-V_{ref}). 35

3.7 The Adaptive weight circuit based on the leaky LMS algorithm. . . . 37

3.8 Absolute-value circuit. 38

3.9 HSPICE simulation of absolute-value circuit. 39

3.10 The measurements of the early effect for L=5 μm and 25 μm. 40

3.11 Gamma filter measured chip results. (a) Bias voltage vs. 3 dB cutoff frequency. (b) Transfer function of first two taps gamma filter. . . . 42
3.12 (a) Simulated and (b) measured impulse response of gamma filter tap outputs. 42
3.13 Circuit to generate the input noise signal for the system identification problem. 43
3.14 Pseudo-random bit stream generated by noise chip. 44
3.15 Power spectral density of noise chip. Top, before filtering. Bottom, after RC filtering. 45
3.16 Second order Sallen-Key circuit 46
3.17 System identification flow chart. 46
3.18 The original filtered pseudo-random bit stream is shown at the top. The lower plot shows the the noise after passing through the discrete Sallen-Key circuit. The adaptive filter must mimic the performance of this circuit. 47
3.19 (a) A sample of the desired and the output signal before convergence. (b) Another sample of the same signals after convergence. 48
3.20 (a) The trajectory of the weight w_0 during convergence for three different initial values. (b) The trajectory for both weights w_0 and w_1. 49
3.21 Weight tracks based on the leaky LMS algorithm. (a) Initial weight voltages equal 3 V. (b) Initial weight voltages equal 2 V. 50
3.22 The absolute instantaneous error signal for the system ID problem. . 50
3.23 Noise cancellation chart. 51
3.24 Top left: signal $s(t)$. Top right: $n_o(t)$. Bottom left: $s(t) + n_o(t)$. Bottom right: $n_1(t)$. 53
3.25 The signal source, $s(t)$, and the system output signal $\varepsilon(t)$ after convergence. 53

4.1 Continuous-time Laguerre kernels. 56
4.2 Continuous-time Laguerre memory structure. 56
4.3 First-order all-pass circuit. 59
4.4 First-order all-pass circuit with active transistors. 60
4.5 The 4-tap Laguerre filter transfer function. (a) HSPICE simulation result. (b) Chip measurement. 61
4.6 Measured results of 4-tap Laguerre filter. 62
4.7 Block diagram for system identification simulation. 62
4.8 The faster learning rate of the Laguerre vs. the gamma. 63
4.9 The comparison of MSE vs. the amount of added random noise for the Laguerre and gamma. The Laguerre shows a lower MSE for the same amount of added noise at the tap outputs. 63
4.10 Weights track with different initial values. (a) Initial weights equal 2.0 V. (b) Initial weights equal 3.0 V. 64
4.11 w_0 weight track comparison: (a) comparison on Laguerre filter with initialized all weights and initialized one weight after stable. (b) comparison weight tracks between the Laguerre and gamma filters. . . . 65

5.1 The generalized feedforward filter. 68
5.2 Continuous-time multi-scale gamma memory. 69
5.3 (a) Simulated multi-scale gamma kernels. (b) Measured kernels. . . . 70
5.4 Impulse responses of the multi-scale gamma memory on a log-time plot. The peak value of each impulse response has been normalized to unity for display purposes. 71
5.5 Circuit implementation of the multi-scale gamma structure. 72
5.6 Performance index comparison of the gamma and multi-scale gamma filters. 73

5.7 Performance index of multi-scale gamma with different numbers of taps (L). 74
5.8 The tap outputs of ms-gamma filter. The first tap output is shown in top left, the second tap output is shown in top right. The 8^{th} tap is shown in bottom right. 75
5.9 The tap outputs of gamma filter with the bias voltage equal 0.65 V. The first tap output is shown in top left, the second tap output is shown in top right. The 8^{th} tap is shown in bottom right. 75
5.10 The tap outputs of gamma filter with the bias voltage equal 0.75 V. The first tap output is shown in top left, the second tap output is shown in top right. The 8^{th} tap is shown in bottom right. 76
5.11 The output signal and desired signals. Top, at $t = 0$. Bottom, after convergence. 77
5.12 Performance index of the ms-gamma and gamma with two different transfer functions. Straight lines show the performance index of ms-gamma filter while error bar curves show the performance of the gamma filter in two different unknown systems. 77
5.13 Multi-scale Laguerre memory structure. 78
5.14 A contour for the evaluation of the integral in the upper-half complex plane. 79
5.15 Multi-scale Laguerre single tap circuit. 81
5.16 Multi-scale Laguerre circuit. The resistors with the same dashed line is biased at the same voltage. 82

6.1 Discrete-time gamma filter. 85
6.2 Discrete-time Laguerre filter. 86
6.3 Discrete-time multi-scale gamma filter. 91

6.4 Mean square error comparisons between the gamma filter (solid line) and the multi-scale gamma filter (dotted line). We plot MSE vs μ (for gamma) or a for (multi-scale gamma). 93
6.5 Plot of MSE vs. number of taps using the weight pruning method. For each number of taps, the smallest magnitude weights are set to zero. 94
6.6 Block diagram for generating the echo. 95
6.7 The ms-gamma performance index respective to μ and a. 96
6.8 (a) MSE comparison for repeated pole echo transfer function, (b) MSE comparison for nonrepeated pole echo transfer function. 97
6.9 (a) The learning curves comparison of 3^{th}-order gamma and 7^{th}-order ms-gamma. (b) The convergence curve of μ for the gamma filter. . . 98

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

ANALOG IMPLEMENTATIONS OF LOCALLY RECURRENT ADAPTIVE FILTERS

By

Jui-Kuo Juan

May 1998

Chairman: Dr. John G. Harris
Major Department: Electrical and Computer Engineering

The discrete-time transversal filter using the ideal delay line has been widely used in adaptive filters. However, the continuous-time transversal filter has been much less studied. Even in the present digital computer era, analog filters possess a definite place due to their excellent performance at high frequencies. The goal of this research is to develop new continuous-time generalized structures which can capture the merits of stability, simple algorithms, and finally implementation on VLSI chip.

The continuous-time implementation of the transversal filter is problematic since it is impossible to implement an ideal time delay in continuous-time hardware. We first build a continuous-time gamma filter which has been shown superior to FIR filters in discrete-time domain. Furthermore, a cascade all-pass filters creates an orthogonal basis called the Laguerre filter that is shown to outperform the gamma filter both on simulations and chip data measurements. The single time constant of these two generalized structures provides the extra freedom to balance the memory depth

and resolution of the filters and is a powerful tool in system identification and noise cancellation. However, the multimodal performance surface of the gamma/Laguerre filters ensures that the optimum solution is difficult to find due to local minima.

Hence, rather than performing gradient descent on a multimodal error function to determine a single optimal time constant, we propose a multi-scale realization of these delay line structures. Both continuous-time chip measurements and discrete-time simulations show that the multi-scale filters can achieve the same performance with a single time scale gamma/Laguerre filters without the adaptation of time constant.

CHAPTER 1
INTRODUCTION

1.1 Motivation for Analog Implementation of Adaptive Filters

Adaptive filters are of fundamental importance in many areas of signal processing. Adaptive systems have been used to solve problems in such area as prediction, system identification, equalization, and interference canceling [49]. Since the advent of digital computers, nearly all research in transversal adaptive filters has concentrated on discrete-time structures. However, even in the present digital computer era, the analog hardware adaptive structures discussed in this thesis are of great research interest because they can solve many real-world problem faster, smaller, cheaper, and with lower power than the digital alternatives. The continuous-time approach must deal with the problems inherent in analog designs (such as component mismatches and long-term storage) and find simple enough structures that can be implemented efficiently on a chip. Among these adaptive systems, we decided to develop and implement the transversal-type filter structures because of their simplicity and success in problems such as system identification, linear prediction, channel equalization and echo cancellation.

1.2 Discrete-time Adaptive Filters

Early adaptive engineering systems have been designed and implemented in the areas of communication and transmission. Actual implementations of adaptive transmission equalizers by Bell Laboratories date back to 1965 [50, 28]. Nearly three decades, the interest of research in the field of adaptive signal processing has focused on discrete-time. An L-tap discrete-time transversal filter, shown in Figure 1.1, using

ideal tap delay lines has been the dominant form of short-term memory-based filtering for adaptive systems. The reason for its success is, besides the simplicity of the transversal filter structure, the unimodality of its error surface, and the existence of fast and efficient adaptive algorithms to adjust its parameters.

The principal problem with the discrete-time transversal filter, which is also

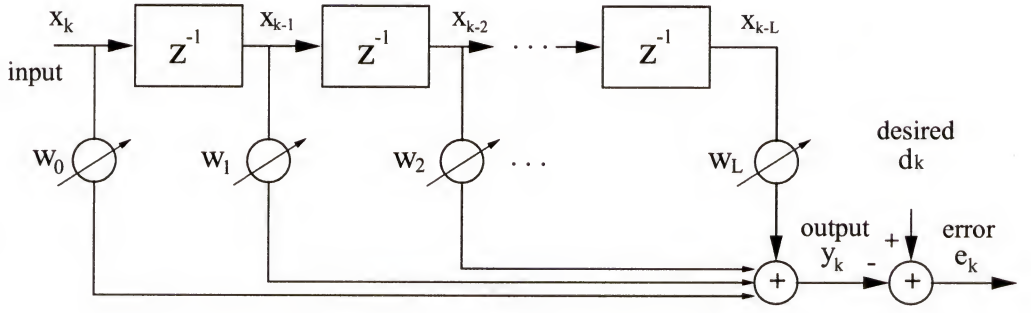


Figure 1.1: Discrete-time transversal filter.

related to its advantage, is that its impulse response has a finite duration (it is a FIR filter). Hence, the memory depth of an FIR filter intimately tied to the number of taps. For this reason, when this filter is used to approximate a system with a long (possibly infinite) impulse response the number of delays of the filter required to provide an acceptable approximation can be quite high. For instance, in order to suppress echo effects on transcontinental calls via satellite links, whose two-way round-trip time delay is nearly half a second, it requires up to 128 taps for the 8 kHz sampling rate for speech [46].

The problem of the FIR adaptive filters can be partially solved using filters with an infinite impulse response (IIR filters), where memory depth is uncoupled with the order of the filter. IIR filters create arbitrary long impulse responses with a small order. However, these filters have their own problems, especially if output error models are used. Among these are likely multi-modal error surfaces, and possible instability problems related to the adaptation of the poles of these filters. Therefore,

choosing a proper filter between FIR and IIR becomes a dilemma that strongly depends on the problems to be solved.

1.3 Continuous-time Generalized Feedforward Filters

It is impossible to realize an all-zero continuous-time filter. However, if we follow the definition of the discrete-time FIR transversal filters, then a continuous-time adaptive transversal filter will require an ideal delay among these taps. Unfortunately, it is impossible to implement an ideal delay line in analog hardware. As a consequence, many hardware systems substitute low-pass filters for ideal delays and their designers may have hoped that they have not sacrificed much in terms of system performance. Ironically, the results of this thesis indicate that nothing is sacrificed but in fact the low-pass filters provide an improvement in performance over the ideal delays (even if they could be built) [41]. Actually, using a low-pass filter to replace an ideal delay in continuous-time transversal-type filter is not the only solution, it turns out searching other possible alternatives is of high interest. A new class of filters known as the generalized feedforward filter was popularized with the development of the gamma filter [41]. The generalized feedforward filter has a structure similar to transversal filter with the ideal delay unit replaced by a more general transfer functions, $H_k(s)$. It turns out the generalized feedforward filter has a reduced complexity compared to IIR filters while offering more capability than FIR filters. This special structure filter can be directly applied in both discrete-time and continuous-time domains.

A continuous-time generalized feedforward filter is shown in Figure 1.2. Compared to the discrete-time transversal filter in Figure 1.1, both filters have similar structures. The ideal tap delay block in the discrete-time is now replaced by a transfer function H_k(s). From the point of view of signal processing, the L-tap continuous-time generalized feedforward filter is a one input/one output system, where the tap

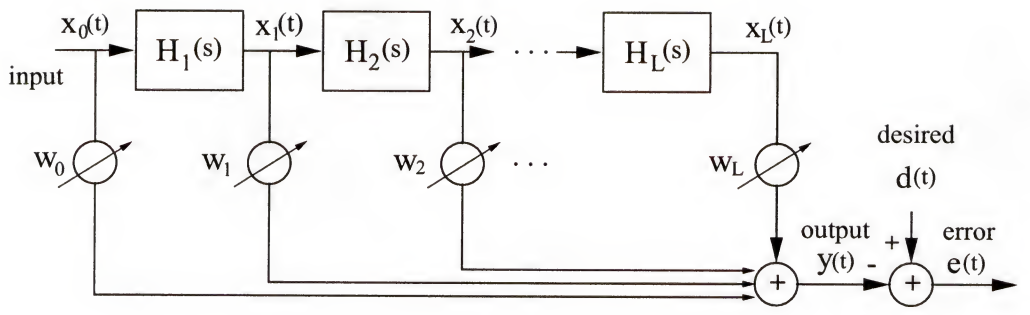


Figure 1.2: Continuous-time generalized feedforward transversal filter.

outputs can be viewed as the projection bases of single input in L-dimension space. These bases are determined by the transfer functions, $H_k(s)$. Jones summarized some popular kernel(bases) including the gamma, Laguerre, Legendre, Jacobi, and Kautz in both discrete and continuous-time domains [14]. In each case, the transfer functions, $H_k(s)$, has the following characteristics:

1. The memory depth is related to H_k(s) as well as the filter order. We can control the length of the impulse response by changing the transfer function H_k(s).

2. The feedback connections are restricted to be internal to the individual H_k(s) blocks. More global feedback connections are prohibited.

3. Although the transfer function $H_k(s)$ can be chosen arbitrary with any order, the $H_k(s)$ is constrained to be first-order in this thesis. Under this restriction, stability can be easily guaranteed and the optimal solution can be computed with less difficulty.

A major part of the dissertation is to analyze and implement some of the generalized feedforward filters mentioned above as well as to develop new ones.

1.4 Sub-threshold Region

Since most of the circuits discussed in this thesis are operated in the CMOS sub-threshold region, it is necessary for us to review the characteristics of MOSFET in this region. Most of the references for sub-threshold implementations can be found in Mead's book [31]. The relation between the input gate voltage and the output drain current for an NMOS transistor can be written as:

$$I = I_o e^{\frac{\kappa V_g}{V_T}} (e^{\frac{-V_s}{V_T}} - e^{\frac{V_d}{V_T}}) \tag{1.1}$$

where V_T is the thermal voltage (about 25 mV at room temperature). The fabrication constants κ and I_o express the effectiveness of the gate in determining the surface potential, and the leakage current. From Equation (1.1), circuits operated in the sub-threshold region provide a couple of advantages. First, since the drain currents are very small, the power dissipation is extremely low. Second, the drain current saturates in a few thermal voltages, roughly 100 mV, which provides a very good characteristic for a current source. Third, the drain current increases exponentially with the gate voltage. The exponential nonlinearity relation extends the dynamic range of the output drain current over many orders of magnitude [31].

With the understanding of the characteristics of sub-threshold NMOS behavior, the multipliers, adders, and transamps can be designed in a similar fashion to bipolar circuit design. The other alternative is to operate the MOSFET in the above-threshold region. For instance, Bult proposed another implementation of analog CMOS circuits based on the square-law characteristic in saturation region [2].

1.5 Previous Work

Analog VLSI signal processing is most effective when high precision is not required, and is therefore an ideal solution for the implementation of perception systems. It is a commonly accepted view that the role of analog in future VLSI circuits and systems will be confined to that of an interface, the very thin "analog shell" between the fully analog outer world and the fully digital substance of the growing signal processing "egg." For perception systems the very precise computation on sequences of numbers is certainly not what is needed, what is needed is a massively parallel collective processing of a large number of a signals that are continuous in time and in amplitude [48]. We will review the previous work of designing an arbitrary analog adaptive IIR filter using the so-called state-space model followed by other implementations of neural networks using analog circuits.

1.5.1 State-space Adaptive Recursive Filters

Although IIR filters are more powerful than FIR filters in modeling arbitrary systems, designing an IIR filter faces potential instability and local minimum problems. Several methods have been proposed to implement arbitrary transfer functions, one of these methods is to introduce state-space concepts. Several papers had proposed using state-space variables, or so called Intermediate Functions (IF) that can realize arbitrary systems [10, 11, 12, 13, 45]. This method can realize a given N^{th} order transfer function, T(s), with a structure of N resistively interconnected integrators. The design process consists of two steps. First, select a set of N functions, called intermediate transfer functions from the given transfer function T(s). Second, synthesize a circuit realization of T(s) from the selected IF. The major advantage of this method is that the IF's can be very effectively employed in determining the expected sensitivity and dynamic range performance of the active filter. Usually an N^{th} order active filter can be described by the state-variable formulation.

$$\begin{aligned} s\mathbf{X}(\mathrm{s}) &= \mathbf{A}\mathbf{X}(\mathrm{s}) + \mathbf{b}\mathrm{U}(\mathrm{s}) \\ Y(s) &= \mathbf{c}^{\mathbf{T}}\mathbf{X}(\mathrm{s}) + \mathrm{dU}(\mathrm{s}) \end{aligned} \tag{1.2}$$

where U(s) is an input signal and $\mathbf{X}(s)$ is the circuit state. $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$, and d are coefficients relating these variables. The $\mathbf{A}$ is a NxN matrix while $\mathbf{b}$ and $\mathbf{c}$ are Nx1 vectors, and d is a scalar. The transfer function, $T(s)$, defined as the ratio of output, $Y(s)$, over input signal, $U(s)$, can be easily derived from Equation (1.2)

$$T(s) = \mathbf{c}^{\mathbf{T}}(\mathbf{sI} - \mathbf{A})^{-\mathbf{1}}\mathbf{b} + \mathrm{d} \tag{1.3}$$

The coefficient update formula for the state-space coefficients, $\mathbf{A}, \mathbf{b}, \mathbf{c}$, d, can be obtained with gradient signals. An approach for adapting all the coefficients is given by [13], where N extra filters are required to obtain all the necessary gradient signals. Unfortunately, these extra filters would require a large silicon area and therefore limit the practicality of this general approach. Furthermore since the presented algorithms are all gradient-based using IIR structures, the issues involving filter instability and global convergence should be solved before these techniques become useful realizations.

1.5.2 Analog Implementation of Neural Networks

There are no known reports in the literature of analog implementation of multi-stage transversal-type filters. However, there have been many papers reporting on analog VLSI implementations of neural networks. For instance, Lont had implemented a multi-layer perceptron with nonlinear synapses. The synapse weights are stored in an off-chip digital memory and loaded to the chip via D/A converter, where they are temporarily stored in DRAM-like capacitor cells. This chip was used to recognize simple patterns, the first four alphabet letters in 4x4 pixel templates [26].

Similar work had been reported by Salam and Oh who implemented a simple 6x2 neural network and demonstrated prediction and system modeling with a single sine wave input signal [43]. An adaptive analog continuous-time biquadratic filter operating at 300 KHz was implemented by Kwan [22]. The biquad filter implemented notch, band-pass, and low-pass transfer function, the only parameter adapted is the resonant frequency and the band-pass center frequency. Moon represented a VLSI implementation of synapse signal weighting and summing using coded neural-type cells (NTC). The NTC is an electronic analogy of a biological soma; it initiated reactions, with a given external voltage (stimulus), by generating a stream of electrical (biochemical) pulse waves, where a threshold level is determined by an inverter logic threshold voltage [32]. Other implementations included a CMOS analog adaptive bidirectional associative memory with on-chip Hebbian learning and on-chip analog weight storage capability reported in [23].

1.6 Organization of this Work

In this dissertation, we discuss the design, implementation, and characterization of several continuous-time generalized feedforward filters. The simulation of discrete-time counterpart is also presented. The text is organized as follow:

• **Chapter 2**: We discuss the properties of continuous-time adaptive transversal-type filters. These results are immediately compared to properties of their discrete-time counterpart which have already been well-studied. First, we show that the Wiener-Hopf solution for both continuous and discrete-time domains possess the same form. We prove there is no upper bound for the learning rate in continuous-time domain. While in discrete-time, the learning rate is always bounded by the eigenvalues of the correlation matrix. We show the convergence time constant is set by the reciprocal of the product of the learning rate and the minimum eigenvalue. For the LMS algorithm, we present a simple example to demonstrate the convergence of the weight

and show the tradeoff between the convergence time constant and the misadjustment.

• **Chapter 3**: This chapter discusses the first-ever analog VLSI implementation of the gamma filter. The gamma filter is the most popular generalized feedforward filter and was proposed by Principe in 1993 [41]. The gamma filter is cascaded by identical first-order low-pass filters, its unique time constant can provide a flexible solution between the tap resolution and filter memory depth. These circuit implementations of the gamma filter are designed to operate on the sub-threshold region for the reasons of extended dynamic range and low power dissipation. We successfully test the gamma filter with the problems of system identification and noise cancellation. We analyze these results and use them for further comparison with other filters.

• **Chapter 4**: The first analog VLSI implementation of the Laguerre filter is presented in the chapter. Unlike the gamma memory, the impulse response at each tap of the Laguerre memory forms an orthogonal basis which is advantageous for representations of signals. The Laguerre filter is a cascade of identical first-order all-pass functions. From the view of circuit design, the amplitude of the signal is not attenuated when propagating along these cascade all-pass filters, only the phase is distorted. These all-pass filters guarantee the signal has the same amplitude (S/N ratio) in each tap while gamma whose amplitude could diminish along these cascade low-pass filters. Simulation results show the Laguerre filter has faster convergence speed than the gamma filter. A Laguerre filter has been built and tested, and measurements agree with simulation results.

• **Chapter 5**: The major problem with the gamma and Laguerre filters in continuous-time or discrete-time is that gradient descent method is not guaranteed to find the optimal time scale during adaptation. This problem becomes particularly troublesome when we build dedicated hardware for implementing the gamma and Laguerre filters. Therefore, we introduce a new generalized feedforward filter called the multi-scale gamma/Laguerre. The multi-scale concept is introduced to avoid the

unpleasant search for the optimal time scale. The ms-gamma provides different time constants depending on the parameter a. We show that each tap output kernel of the ms-gamma has the same shape except for the amplitude and time scale. Moreover, we also prove that the ms-Laguerre kernels we propose are orthogonal. The simulations show the ms-gamma has the same performance for our chosen problems even without the adaptation of time scale. Further, for system identification problem, a ms-gamma chip demonstrates its performances is compatible to gamma filter with different systems.

• **Chapter 6**: We believe that the multi-scale concept is interesting in discrete-time implementation as well as in continuous-time. For this reason, this chapter discusses the discrete-time implementation of the multi-scale gamma. We compare the results between the ms-gamma and gamma filter in different problems. These results imply that the ms-gamma can achieve the same performance as the gamma in different problems.

• **Chapter 7**: We conclude by showing how we have met the goals of this research, by pointing to several drawbacks of the present design, and by outlining further directions of research.

• **Appendix A**: We show the derivation of the DC offsets on the LMS algorithm.

CHAPTER 2
CONTINUOUS-TIME ADAPTIVE FILTER

This chapter discusses continuous-time transversal adaptive filters. The continuous time adaptive filters were first investigated by Wiener in the 1940's [25], but have developed little since that time. In this chapter, we will develop the properties in continuous-time which are well known in the discrete-time domain. These issues include the proper learning rate, convergence time, and the misadjustment. The misadjustment provides a measure of the difference between actual and optimal performance averaged over time. First, we derive the Wiener-Hopf equation for continuous-time generalized feedforward filters. We shall see that the Wiener solution result has the same form in both the continuous and the discrete domains. Later, the learning rate not only controls the convergence speed of the filter but also determines the misadjustment. For discrete-time, there is always an upper bound for the learning rate to guarantee stability, which depends on the input signal statistical characteristics. Surprisingly, we show there is no upper bound for the learning rate in the continuous-time domain. Theoretically, the learning rate can be chosen as large as possible to increase the convergence speed, but the misadjustment also increases as the learning rate increases. Furthermore, we discuss the convergence time constant for the discrete and continuous-time domains. Finally, the LMS algorithm is quite useful for practical problems. We show the adaptation rule for the weights and give a example to show the tradeoff between the learning rate and the misadjustment.

2.1 Wiener-Hopf Solution

For the discrete-time transversal adaptive filter shown in Figure 1.1, the memory depth is coupled to the order and the delay between the adjacent taps is simply determined by the sampling rate, T_s. Each tap output is an ideal delay of the previous tap. For an L-tap filter, the output signal y_k

$$y_k = \mathbf{X}_k^T \mathbf{W}_k \tag{2.1}$$

where $\mathbf{X}_k = [x_{0k}\ x_{1k}\ ...\ x_{Lk}]^T$, and $\mathbf{W}_k = [w_{0k}\ w_{1k}\ ...\ w_{Lk}]^T$ are L+1 column vectors. [1] The discrete-time adaptive filter also can be viewed as an FIR filter whose transfer function H_d is

$$H_d(z) = \sum_{n=0}^{L} w_n z^{-n} \tag{2.2}$$

In contrast, an ideal continuous-time delay transversal filter can be realized by

$$\begin{aligned} X_n(s) &= X_{n-1}(s)e^{-sT_s} \qquad in\ s-domain \\ x_n(t) &= x_{n-1}(t-T_s) \qquad in\ time\ domain \end{aligned} \tag{2.3}$$

Each tap signal is a delayed version of the previous tap by time T_s. However, it is impossible to implement such an ideal delay filter. If we use the bilinear transform [47, 33], then

$$z^{-1} = \frac{s-p}{s+p}, \qquad \text{where } p = 2/T_s \tag{2.4}$$

where p is the 3 dB cutoff frequency and T_s is the sampling rate. By substituting Equation (2.4) into (2.2), we show that the memory depth of an ideal continuous-time adaptive filter is approximately the same as that of the discrete-time adaptive filter by using L cascaded first-order all-pass filters. Therefore the transfer function of an analog transversal filter, H(s), can be written as:

$$H(s) = \sum_{n=0}^{L} w_n \left(\frac{s-p}{s+p}\right)^{-n} \tag{2.5}$$

[1] Subscript k shows the sequential samples for the discrete-time case, while n represents the tap number of the filter.

Then, the output signal is given by

$$y(t) = \sum_{n=0}^{L} w_n x_n(t) \tag{2.6}$$

where $x_n(t)$ is given by:

$$\begin{aligned} x_n(t) &= x_{n-1}(t) * L^{-1}\{\frac{s-p}{s+p}\} \\ &= x_{n-1}(t) - 2p\int_0^{\infty} e^{-p\tau} x_{n-1}(t-\tau)d\tau, \qquad n = 1, 2, .., L \end{aligned} \tag{2.7}$$

That means that the tap outputs are not ideal delays of the input signal, but rather are a convolution between input and these all-pass functions. This analog transversal filter is shown in Figure 2.1, where $x_0(t)$ is the input signal.

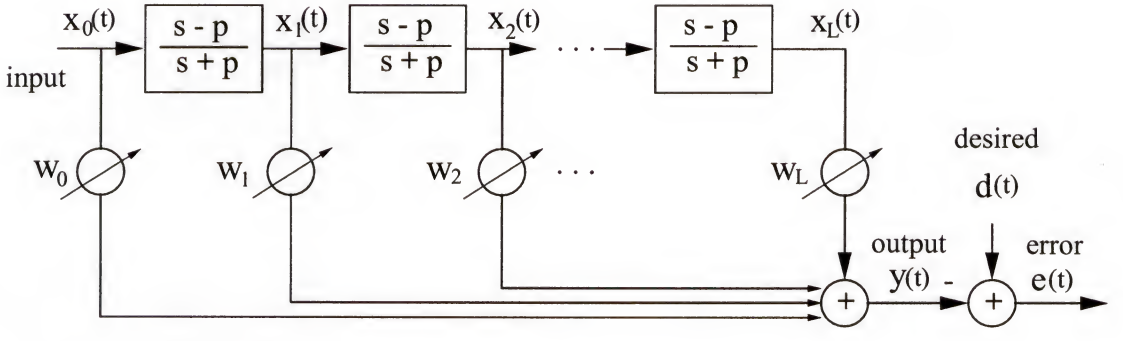


Figure 2.1: An approximated continuous-time ideal delay transversal filter using bilinear transform.

For our goal, we are not interested in approximating an ideal delay line adaptive filter in the continuous-time domain. Instead, we will study the generalized feedforward filters we mention in Chapter 1, because they are much more powerful than the simple delay approximation. This continuous-time filter can be the gamma, Laguerre, Legendre, or any number of different filtering functions.

Suppose the generalized feedforward impulse response is given by $h_n(t)$, then $x_n(t)$ is given by:

$$x_n(t) = h_n(t) * x_{n-1}(t) \tag{2.8}$$

We define the instantaneous error, $e(t)$, as the difference between the desired signal, $d(t)$, and the filter output, $y(t)$,

$$e(t) = d(t) - y(t) \tag{2.9}$$

Squaring the error signal, we obtain the instantaneous squared error:

$$\begin{aligned} e^2(t) &= (d(t) - y(t))^2 \\ &= (d(t) - \mathbf{W}_c^T \mathbf{X}_c(t))^2 \end{aligned} \tag{2.10}$$

The subscript 'c' represents a continuous-time signal, while subscript 'd' is for discrete-time signals. Assuming that $e(t)$, $d(t)$, and $\mathbf{X}_c$ are statistically stationary and take the expected value of Equation (2.10), then

$$\xi_c = E[d(t)^2] - 2\mathbf{P}_c^T \mathbf{W}_c + \mathbf{W}_c^T \mathbf{R}_c \mathbf{W}_c \tag{2.11}$$

where

$$\begin{aligned} \mathbf{R}_c &= E[\mathbf{X}_c(t)\mathbf{X}_c^T(t)] \\ \mathbf{P}_c &= E[d(t)\mathbf{X}_c(t)] \end{aligned} \tag{2.12}$$

The E[·] is the statistical mean of [·]. $\mathbf{R}_c$ is the auto-correlation matrix and $\mathbf{P}_c$ is the cross-correlation vector.

The minimum of ξ_c can be sought by setting the gradient of the mean-square-error with respecting to the weight vector $\mathbf{W}_c$ to zero.

$$\nabla w \xi_c = \frac{\partial \xi_c}{\partial \mathbf{W_c}} = 2\mathbf{R}_c \mathbf{W}_c - 2\mathbf{P_c} = \mathbf{0} \tag{2.13}$$

The Wiener solution, $\mathbf{W}_c^*$, for the continuous-time transversal filter can be solved from Equation (2.13) as:

$$\mathbf{W}_c^* = \mathbf{R}_c^{-1} \mathbf{P}_c \tag{2.14}$$

This result has the same form as in the discrete-time case [49]. Note, this result is only valid when the input signal is stationary.

2.2 The Learning Rate and Convergence Time Constant

We need to consider adaptive filters for nonstationary signals or when the $\mathbf{R}_c^{-1}\mathbf{P}_c$ is too difficult to compute. Let's consider setting the weights using the steepest descent method of ξ_c and assume stationary input signals. For a continuous-time filter, when t approaches infinity, the steady-state solution of the steepest descent method will converge to its Wiener solution, $\mathbf{W}_c^*$. For this method, the weights are adjusted in the direction of the *instantaneous* gradient which can be represented by the following differential equation

$$\frac{d\mathbf{W}(t)}{dt} = -\frac{\rho}{2}\nabla_W \xi_c = -\rho\mathbf{R}_c\mathbf{W}_c(t) + \rho\mathbf{P}_c \tag{2.15}$$

where ρ is the continuous-time learning rate. By solving Equation (2.15), the weight vector at time t can be written as:

$$\mathbf{W}(t) = \mathbf{R}_c^{-1}\mathbf{P}_c + e^{-\rho\mathbf{R}_c t}(\mathbf{W}_c(0) - \mathbf{R}_c^{-1}\mathbf{P}_c) \tag{2.16}$$

where $\mathbf{W}_c(0)$ is the vector of initial weight values. The weight vector is a sum of two terms, since $\mathbf{R}_c$ is positive definite, the exponential term will vanish as t approaches infinity regardless of the ρ and the initial weight values. Hence, as long as the condition $\rho > 0$ is satisfied, the weight vectors will always converge to the Wiener solution. It shows there is no upper bound for learning rate, ρ [51]. This is in sharp contract to the discrete-time case where the learning rate is bound by the eigenvalues of $\mathbf{R}_d$ to ensure stability. In discrete-time the learning rate must be less than the reciprocal of the maximum eigenvalue of $\mathbf{R}_d$ [49]. This result also means that the system can converge arbitrarily fast as long as $\mathbf{R}_c$ is exactly known.

Since $\mathbf{R}_c$ is not a diagonal matrix it must be diagonalized in order to solve the individual weights. By introducing the vector $\mathbf{V}_c'(t)$ and $\mathbf{V}_c(t)$:

$$\begin{aligned} \mathbf{V}_c(t) &= \mathbf{W}_c(t) - \mathbf{W}^*{}_c && \textit{Translation} \\ \mathbf{V}_c'(t) &= \mathbf{Q}_c^T\mathbf{V}_c(t) = \mathbf{Q}_c^{-1}\mathbf{V}_c(t) && \textit{Rotation} \end{aligned} \tag{2.17}$$

where $\mathbf{Q}_c$ is eigenvector matrix of $\mathbf{R}_c$. The $\mathbf{R}_c$ can be decomposed as:

$$\mathbf{R}_c = \mathbf{Q}_c \Lambda_c \mathbf{Q}_c^{-1} \tag{2.18}$$

where Λ_c is a diagonal matrix of the eigenvalues. Substituting these vectors into (2.15), we get

$$\frac{d\mathbf{V}'(t)}{dt} = \mathbf{V}'(0) e^{-\rho \Lambda_c t} \tag{2.19}$$

From Equation (2.19), we observe that each individual convergence speed of the weights depends on ρ as well as its correspondence eigenvalue. Equation (2.19) shows that the individual weight exponentially converges with a time constant, $\tau_{c,n}$, given by:

$$\tau_{c,n} = \frac{1}{\rho \lambda_n} \tag{2.20}$$

where λ_n is n^{th} eigenvalue of Λ_c. In the worst case, the convergence time constant is determined by the minimum eigenvalue, or

$$\tau_c = \frac{1}{\rho \lambda_{c,min}} \tag{2.21}$$

This has the same form as in the discrete-time case where $\tau_d \approx \frac{1}{2\eta\lambda_{d,min}}$. In here $\lambda_{d,min}$ is the minimum eigenvalue of the $\mathbf{R}_d$, and η is the discrete-time learning rate.

In this section, we showed that there is no upper bound for the learning rate of continuous-time adaptive filters using gradient descent method. In discrete-time an upper bound is set by the maximum eigenvalue of $\mathbf{R}_d$. We also showed in both discrete and continuous-time cases, the convergence time constant is set by the reciprocal of the product of the learning rate and the minimum eigenvalue.

2.3 The LMS Algorithm

Usually, the exact statistical characteristics of $\mathbf{R}_c$, $\mathbf{P}_c$, and the gradient are not available. The LMS algorithm uses instantaneous approximation of the auto-correlation matrix and cross-correlation vectors called $\mathbf{R}(t)$, $\mathbf{P}(t)$, i.e., both are functions of time [51].

$$\begin{aligned} \mathbf{P}(t) &= d(t)\mathbf{X}(t) \\ \mathbf{R}(t) &= \mathbf{X}(t)\mathbf{X}^T(t) \end{aligned} \tag{2.22}$$

Replacing the $\mathbf{R}_c$ and $\mathbf{P}_c$ in Equation (2.15) by $\mathbf{R}(t)$ and $\mathbf{P}(t)$, then the differential equation becomes

$$\begin{aligned} \frac{d\mathbf{W}(t)}{dt} &= -\rho\mathbf{R}(t)\mathbf{W}(t) + \rho\mathbf{P}(t) \\ &= \rho[d(t) - \mathbf{X}^T(t)\mathbf{W}(t)]\mathbf{X}(t) \\ &= \rho e(t)\mathbf{X}(t) \end{aligned} \tag{2.23}$$

where Equation (2.22) is used. This is the continuous-time LMS algorithm. Another way to derive continuous-time LMS is to compute the gradient of the instantaneous square error as:

$$\begin{aligned} \frac{d\mathbf{W}(t)}{dt} &= -\frac{\rho}{2}\nabla_W e^2(t) \\ &= \rho e(t)\mathbf{X}(t) \end{aligned} \tag{2.24}$$

We choose a simple example to demonstrate how the continuous-time LMS algorithm works. Consider a single weight ($w_0(t)$) adaptive filter, with an input signal $x(t)$ and a desired signal, d(t)

$$\begin{aligned} x(t) &= A\cos(\omega t) \\ d(t) &= B\cos(\omega t) \end{aligned} \tag{2.25}$$

where A and B are constant amplitudes. From Equation (2.23), the LMS weight update is

$$\frac{dw_0(t)}{dt} = \rho[d(t) - w_0(t)x(t)]x(t) \tag{2.26}$$

Substituting Equation (2.25) into (2.26), we have

$$\frac{dw_0(t)}{dt} = -\frac{A^2}{2}[1 + cos(2\omega t)]w_0(t) + \frac{AB}{2}[1 + cos(2\omega t)] \tag{2.27}$$

Solving the differential equation, the $w_0(t)$ becomes:

$$w_0(t) = \frac{B}{A} - [\frac{B}{A} - w_0(0)]e^{\frac{-A^2\rho}{2}(t+\frac{sin(2\omega t)}{2\omega})} \tag{2.28}$$

as $t \to \infty$

$$w_0(t \to \infty) = \frac{B}{A} \tag{2.29}$$

which is the Wiener solution for this problem. Figure 2.2 shows the convergence of w_0 with two different time rates, ρ. The w_0 converges to $\frac{B}{A}$ regardless the learning rate. The reason is because the desired signal can be exactly represented by the product of the input signal and the optimal weight. i.e., the minimum mean-square-error is equal to 0 in this case.

Furthermore, we assume

$$d(t) = Bcos(\omega t + \phi) \tag{2.30}$$

where ϕ is the phase difference between the input signal and the desired signal. Now, there is no optimal w_0 that can achieve zero mean square error. Figure 2.3 shows the convergence of w_0 with two different learning rates. At steady state, the large learning rate shows a large variance of the weight which infers to a large misadjustment in this case. This result demonstrates the tradeoff between the convergence speed and the misadjustment. In summary, although a large learning rate will not cause divergent in the continuous-time transversal filter, increasing the learning rate will degenerate the performance of the filter.

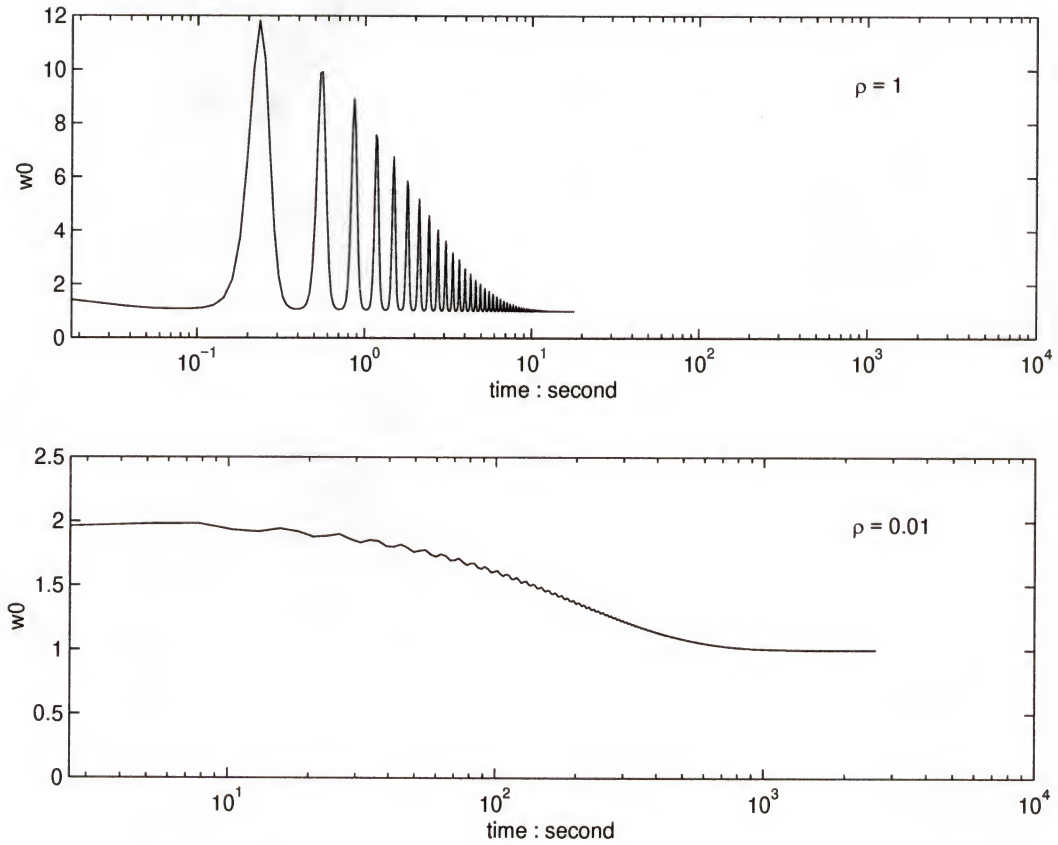


Figure 2.2: Adaptation of w_0. (a) Top, $\rho = 1$. (b) Bottom, $\rho = 0.01$.

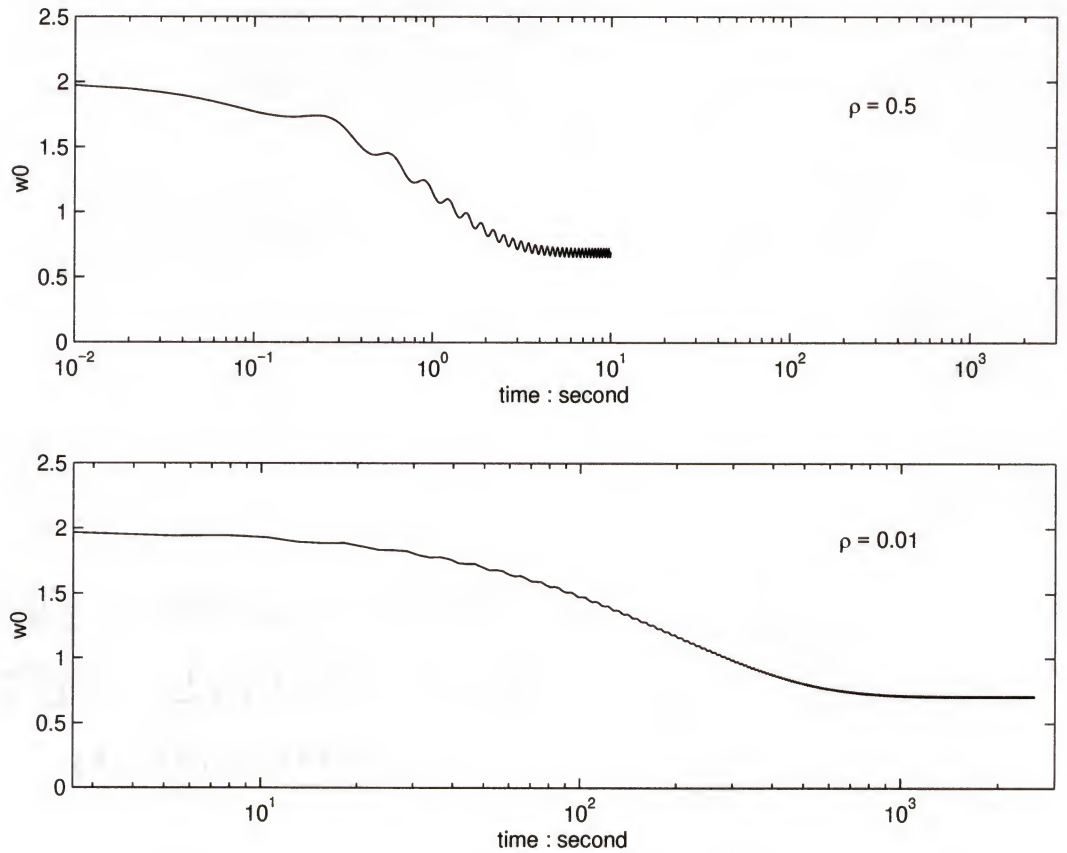


Figure 2.3: Adaptation of w_0. (a) Top, $\rho = 0.5$. (b) Bottom, $\rho = 0.01$. These results show that the misadjustment is increasing as the learning rate is increasing.

CHAPTER 3
CONTINUOUS-TIME GAMMA FILTER

We discuss the continuous-time gamma filter in this chapter. The gamma filter, first proposed by Principe in 1993 [41], is the most popular generalized feedforward filter because of its trivial stability condition, simplicity, and reduced complexity compared to a general IIR filter. The gamma filter has shown success in various area such as linear prediction, system identification, and echo cancellation [36, 4]. Nearly all of this work was done in the discrete-time domain, however, our interest is in the continuous-time gamma filter. We review the characteristics of the gamma kernel and the gamma filter. We also discuss the DC offset problem for the gamma filters and we show both the LMS and Leaky LMS versions. The circuit implementation of the gamma filter includes the gamma transfer function, multiplier, current mirror, and adaptive circuit. All of these circuits were tested in the below-threshold range but can also run above threshold. We test the gamma chip for system identification and noise cancellation problems. These results are discussed in the last section.

3.1 The Gamma Kernel and Gamma Filter

The continuous-time gamma kernels, $g_k(t)$, are defined as:

$$g_k(t) = \frac{1}{\tau(k-1)!}\left(\frac{t}{\tau}\right)^{k-1} e^{-t/\tau} \qquad k = 1, 2, 3, ... \tag{3.1}$$

where τ is the time constant and is therefore a positive number. The name *gamma* is used because $g_k(\mathrm{t})$ is identical to the integrand of the Γ-function, which is defined as $\Gamma\ (\mathrm{x}) = \int_0^\infty t^{x-1}e^{-t}dt$. Furthermore, the area of each gamma kernel is normalized to unity by the factor $\frac{1}{\tau(k-1)!}$. Thus,

$$\int_0^\infty g_k(t)dt = 1 \qquad k = 1, 2, 3, ... \tag{3.2}$$

If we take the derivative of g_k(t) and set it to zero, then the t_k represents the time of the peak of the k^{th} kernel;

$$\begin{aligned} \frac{dg_k(t)}{dt} &= 0 = (k-1) - \frac{t_k}{\tau} \\ t_k &= (k-1)\tau \end{aligned} \tag{3.3}$$

This result shows that t_k is a linear function of k; that is, the peak values of the impulse response are equally spaced by τ for the gamma kernels. Therefore the larger the τ we choose, the longer the time constant we achieve. In other words, the memory depth is extended. Figure 3.1 shows the first four gamma kernels with τ=1 second. As expected, the peak values are reached after 1, 2, and 3 seconds. Note that, $g_k(0) = 0$ for $k > 1$.

We can also explore the frequency characteristics of the gamma kernels. The

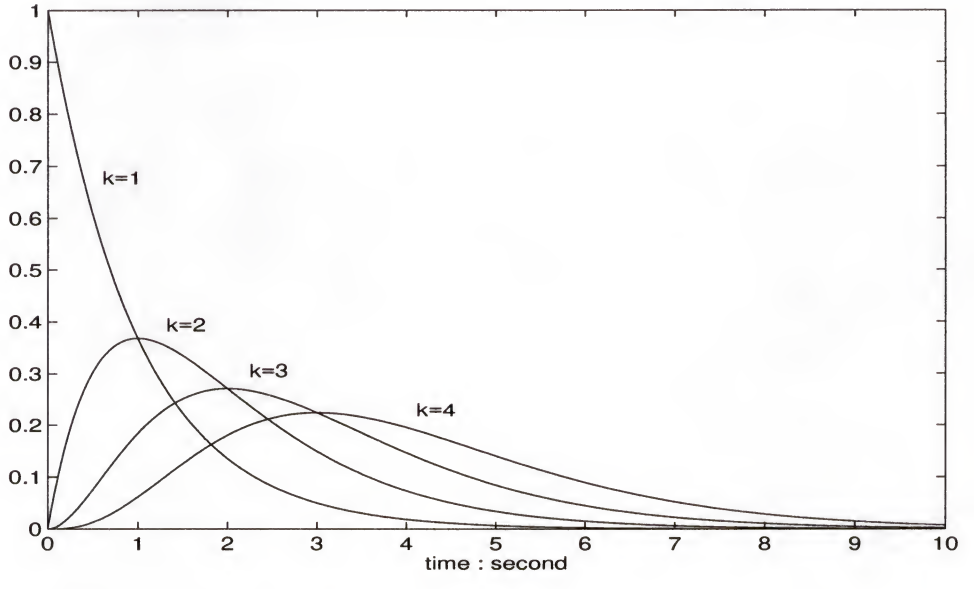


Figure 3.1: Continuous-time gamma kernels.

Laplace transform of the first gamma kernel, $g_1(t)$, becomes

$$G_1(s) = \mathcal{L}[g_1(t)] = \frac{1}{\tau s + 1} = G(s) \tag{3.4}$$

The G(s) represents the gamma unit transfer function which is a first-order low-pass filter. The transform of the higher order gamma kernels are:

$$G_k(s) = G_{k-1}(s)G(s) = (\frac{1}{\tau s + 1})^k \tag{3.5}$$

Equation (3.5) shows that the k^{th} kernel can always be represented by the previous kernel multiplied by the gamma unit transfer function. It is obvious that these kernels have identical poles at $1/\tau$. To construct a gamma memory structure from the generalized feedforward filter shown in Figure 1.2, we can cascade L identical G(s) stages as

$$H_1(s) = H_2(s) = \cdots = H_L(s) = G(s) \tag{3.6}$$

The continuous-time gamma memory structure now can be constructed as shown in Figure 3.2.

Unlike FIR filters, whose memory depth is coupled to its order, the memory

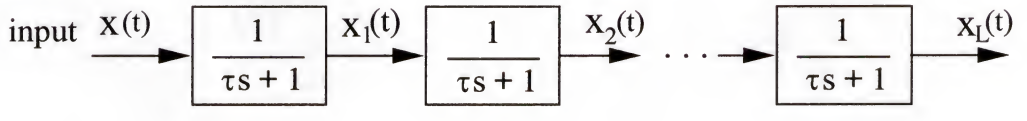


Figure 3.2: Continuous-time gamma memory structure.

depth of the gamma filter depends on τ. We define the memory depth D_m in an L-tap gamma filter as the first moment of the gamma impulse response

$$D_m = \int_0^\infty t \cdot g_L(t) = -\frac{dG_L(s)}{ds}|_{s=0} = L \cdot \tau \tag{3.7}$$

Equation (3.7) shows that the memory depth of gamma filter is proportional to its order as well as the time constant, τ. Further, we can define memory resolution, R_m, as the number of free parameters (i.e., the number of tap variables) per unit of time

in the filter memory

$$R_m = \frac{L}{D_m} \tag{3.8}$$

Now the relation between R_m and D_m of the gamma structure can be expressed as:

$$L = D_m \cdot R_m \tag{3.9}$$

This expression shows that for fixed L (filter order), increasing the memory resolution will also decrease the memory depth. Therefore the time constant,τ, can be used to setting the optimal values for both memory resolution and memory depth.

The Wiener-Hopf equation of the gamma filter can be derived using the same technique mentioned in section 2.1. However, there is one more parameter, τ, that needs to be determined. Rewriting Equation (2.13), the instantaneous squared error, ξ_c

$$\xi_c = \left(d(t) - \mathbf{W}_c^T \mathbf{X}(t)\right)^2 \tag{3.10}$$

Partial differentiation of Equation (3.10) with respect to the weight vector and τ gives:

$$\mathbf{R}_c \mathbf{W}_c = \mathbf{P}_c \tag{3.11}$$

and

$$\mathbf{W}_c^T [\mathbf{R}_\tau \mathbf{W}_c - \mathbf{P}_\tau] = 0 \tag{3.12}$$

where

$$\begin{aligned} \mathbf{R}_\tau &= \frac{\partial \mathbf{R}}{\partial \tau} = E[\mathbf{X}(t) \frac{\partial \mathbf{X}(t)}{\partial \tau}] \\ \mathbf{P}_\tau &= \frac{\partial \mathbf{P}}{\partial \tau} = E[d(t) \frac{\partial \mathbf{X}(t)}{\partial \tau}] \end{aligned} \tag{3.13}$$

The optimal weight vector and optimal τ can be solved by Equation (3.11) and (3.12), respectively.

Finally, since the gamma filter is a special type of IIR, it is interesting to

discuss the state-space model of the gamma filter. The gamma filter state-space model can be written as:

$$\begin{aligned} \dot{\boldsymbol{X}} &= \mathbf{A}\boldsymbol{X} + \mathbf{b}\mathrm{U} \\ Y &= \mathbf{c}^{\mathbf{T}}\boldsymbol{X} + \mathrm{dU} \end{aligned} \tag{3.14}$$

Now the tap outputs are represented by a new state-variable and U(t) is the input signal. First we can re-write equation (3.4) as:

$$s\mathbf{X}_{k+1}(\mathbf{s}) = \frac{\mathbf{1}}{\tau}[\mathbf{X}_k(\mathbf{s}) - \mathbf{X}_{k+1}(\mathbf{s})] \tag{3.15}$$

The equation can be transformed to the time domain by

$$\frac{dx_{k+1}(t)}{dt} = \frac{1}{\tau}[x_k(t) - x_{k+1}(t)] \tag{3.16}$$

then the overall state-space equations become

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ \cdot \\ \cdot \\ x_k(t) \end{bmatrix} = \begin{bmatrix} -1/\tau & 0 & 0 & 0 \\ 1/\tau & -1/\tau & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 1/\tau & -1/\tau \end{bmatrix} \begin{bmatrix} x_1(t) \\ \cdot \\ \cdot \\ x_k(t) \end{bmatrix} + \begin{bmatrix} 1/\tau \\ 0 \\ 0 \\ 0 \end{bmatrix} U(t) \tag{3.17}$$

On the other hand, the output signal equation is a linear combiner of tap outputs and weights

$$y(t) = \begin{bmatrix} w_1 & \cdot & \cdot & w_k \end{bmatrix} \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_k \end{bmatrix} + w_0 U(t) \tag{3.18}$$

Comparing Equations (3.17) and (3.18) with (3.14), the coefficients of state-space gamma filter can be obtained.

3.2 The LMS Algorithm

We have mentioned the LMS algorithm in Chapter 2, however, we still need to derive the update rule for the time constant, τ, in the gamma filter based on

the LMS algorithm. Further, a modified LMS called leaky LMS will be discussed in Section 3.4. The idea behind LMS is try to force the weight update vector to move along a line in the weight space parallel to its gradient signal vector. Actually, the weight updates move on average in the direction of steepest decent of the squared error surface. By treating the tap outputs as the fixed delay signal during the update weight, the LMS algorithm is well suited for the gamma filter. First, the output signal is represented as a linear combiner of the of weights and tap outputs. If $y(t)$ is used to represent the output signal, then

$$y(t) = \sum_{k=0}^{L} w_k(t) \cdot x_k(t) \tag{3.19}$$

where L is the order of the gamma filter. The error signal, $e(t)$, is simply equal to the difference between the desired signal and the output signal

$$e(t) = d(t) - y(t) \tag{3.20}$$

Now the instantaneous cost function $E(t)$ can be written as:

$$E(t) = \frac{1}{2}e^2(t) = \frac{1}{2}[d(t) - y(t)]^2 \tag{3.21}$$

The weight difference $\Delta w_k(t)$ in a small time interval can be expressed as:

$$\Delta w_k(t) = -\eta_w \Delta t \frac{\partial E}{\partial w_k} = \eta_w \Delta t e(t) x_k(t) \tag{3.22}$$

where η_w is the learning rate of the weights. In the limit as Δt approaches to zero, Equation (3.22) becomes a differential equation

$$\frac{dw_k(t)}{dt} = \eta_w e(t) x_k(t) \tag{3.23}$$

Furthermore, the update for time constant, τ, will become more complicated because it involves all of the tap outputs. First, let $\mu = 1/\tau$ and use the same strategy as was done with the weights, the change of μ in a small interval Δt follows the rule:

$$\begin{aligned} \Delta\mu(t) &= -\eta_\mu \frac{\partial E}{\partial \mu} \\ &= \eta_\mu e(t) \Delta t \sum_{k=0}^{L} w_k(t) \frac{\partial x_k(t)}{\partial \mu} \end{aligned} \tag{3.24}$$

the term of $\frac{\partial x_k(t)}{\partial \mu}$ can be solved by using the gamma kernel to replace tap output $x_k(t)$, then

$$\frac{\partial x_k(t)}{\partial \mu} = \frac{k}{\mu}[x_k(t) - x_{k+1}(t)] \tag{3.25}$$

By substituting this result into Equation (3.24), and letting Δt approach zero, the update rule for μ becomes:

$$\frac{d\mu(t)}{dt} = \frac{\eta_\mu}{\mu(t)} e(t) \sum_{k=0}^{L} k w_k(t)[x_k(t) - x_{k+1}(t)] \tag{3.26}$$

The update of μ involves every tap output from 0 to L plus an extra tap, L+1. Apparently, updating weights is much easier than updating time constant because only individual tap outputs are needed. A similar update rule for μ in discrete-time domain was proposed in [36, 37].

3.3 DC Offset Effects in the LMS Algorithm

The LMS algorithm provides a powerful tool for adaptive filters but their analog implementations pose a few problems such the DC offset problem. DC offsets are not relevant for digital implementation, but are a major problem for analog implementations. For analog adaptive filters, it is well known that DC offsets present in the LMS algorithm can affect system performance [44, 6]. The DC offsets cause the filter weights to be incorrect resulting in an error in the programmable filter is transfer function at all frequencies. A detailed analysis of the DC offset effect in the generalized feedforward filter can be found in Appendix A.

Appendix A shows a derivation of the effect of additive offsets in each tap output ($\mathbf{m_x}$) and an offset in the error calculation (m_e). The final result is that the excess variance of the resulting error caused by the DC offset is given by:

$$\sigma_e^2 \simeq (\mathbf{m_x} \mathrm{m_e})^{\mathrm{T}} \mathrm{R}^{-\mathrm{T}} (\mathbf{m_x} \mathrm{m_e}) \tag{3.27}$$

The result shows the variance of error caused by DC offsets is not only determined by the magnitudes of $\mathbf{m_x}$ and m_e but also depends on the auto-correlation of the

input signal. Besides, the excess MSE is independent on η_w. The excess MSE due to offset can be reduced by using a higher input signal power. However, it is impossible to avoid these DC offsets $\mathbf{m_x}$, m_e, which mainly depend on the layout. Methods to reduce these offsets are possible with well-designed symmetric transamp circuits and careful layouts. These circuits will be addressed later in this thesis.

3.4 The Leaky LMS Algorithm

When the LMS algorithm is used, DC offsets may prevent the weights from reaching correct steady-state values and may even cause divergence. Several modifications to the LMS algorithm have been proposed to solve these stability problems. The leaky LMS algorithm is successful in alleviating "stalling" where the gradient estimate is too small to adjust the coefficients of the algorithm due to a very low input signal. The leaky LMS provides a compromise between minimizing the mean-square error and minimizing the magnitude of the weights [9, 49]. The stability factor is attained at the expense of an increase in hardware cost and worse performance compared to the conventional LMS algorithm.

The leaky algorithm is introduced to minimize the instantaneous cost function $E(t)$

$$E(t) = \frac{1}{2}e^2(t) + \gamma W^T(t)W(t) \tag{3.28}$$

The cost function is similar to Equation (3.21) except with an extra term $\gamma W^T(t)W(t)$ which is controlled by the leaky factor γ, where $0 \leq \gamma \leq 1$. When $\gamma = 0$ the leaky algorithm reduces to the conventional LMS. Now the update rule of $w_k(t)$ can be derived using the same method as that used for LMS:

$$\frac{dw_k(t)}{dt} = \eta_w e(t)x_k(t) - \eta_w \gamma w_k(t) \tag{3.29}$$

The leakage factor γ allows the filter weight vector to decay with time. In steady state, if the error signal is equal to zero, then the weight vector will decrease to zero.

3.5 Circuit Implementations

To implement the gamma filter with the leaky LMS algorithm, we need the gamma transfer function (first-order low-pass filter), the adders, multipliers, and current mirrors. These circuits can be efficiency implemented in sub-threshold region.

3.5.1 Gamma Transfer Function

A modified transconductance amplifier is shown in Figure 3.3(a). The bias voltage is always below the threshold voltage. Comparing to original transamp, the extra diode pair added below the input/output (acts like emitter degeneration) is used to reduce the transconductance by half, therefore we can double the input linear range.

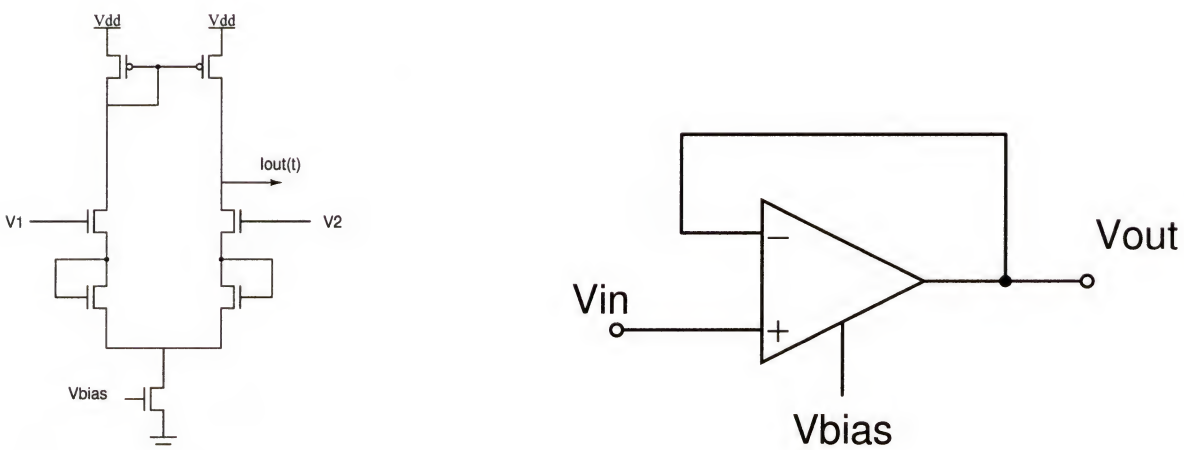


Figure 3.3: (a) Modified transconductance amplifier. (b) Symbol of transamp.

The output current of the transamp, I_{out}, can be written as:

$$I_{out} = I_b \tanh[\frac{\kappa(V_1 - V_2)}{4V_T}] \tag{3.30}$$

If we connect the transconductance amplifier as a follower-integrator, the derivative of the capacitor's voltage is proportional to the output current of the transamp.

$$C\frac{dV_{out}(t)}{dt} = I_b \tanh[\frac{\kappa(V_{in}(t) - Vout(t))}{4V_T}] \tag{3.31}$$

For small signals, the tanh can be approximated by its argument. Assuming the thermal voltage is 25 mV in room temperature, then if the input voltage magnitude is constrained under 200 mV peak to peak, the equation can be approximately linearized to

$$C\frac{dV_{out}(t)}{dt} = G(V_{in}(t) - V_{out}(t)) \tag{3.32}$$

where G, the transconductance, is equal to

$$G = \frac{\partial I_{out}}{\partial V_{in}} = \frac{I_b}{4V_T/\kappa} \tag{3.33}$$

In the s-domain, the transfer function of the transamp can be derived from Equation (3.32) as:

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{\tau s + 1} = \frac{\mu}{s + \mu} \tag{3.34}$$

The transfer function of the transamp is identical to the gamma transfer function, G(s). Here, μ = G/C is the 3 dB cutoff frequency of the low-pass filter. The CMOS transamp is operated in the sub-threshold region so that a large dynamic range of τ can be obtained. For speech processing applications, the necessary dynamic range of $2\pi/\tau$ is from 100 Hz to 10 KHz, which can be achieved with the exponentially controlled bias voltage. On the other hand, since the input impedance of the transamp is nearly infinite, we can simply cascade L transamps to construct a L-tap gamma memory without any degradation. For a four tap gamma memory, the comparison of the DC offsets, $\mathbf{m_x}$, simulated by HSPICE (setting the input voltage equaling to 2.5 V) is summarized in Table 3.1. Note, Table 3.1 represents only the systematic offset since the random offsets are not easily modeled in HSPICE. Figure 3.4 shows the linear ranges of the transamp with/without extra diode pairs.

Table 3.1: Comparison on DC offset $\mathbf{m_x}$ unit : mV

transamp	tap 1	tap 2	tap 3	tap 4
without diodes	9.8	19.6	29.1	38.7
with diodes	0.6	1.1	1.7	2.3

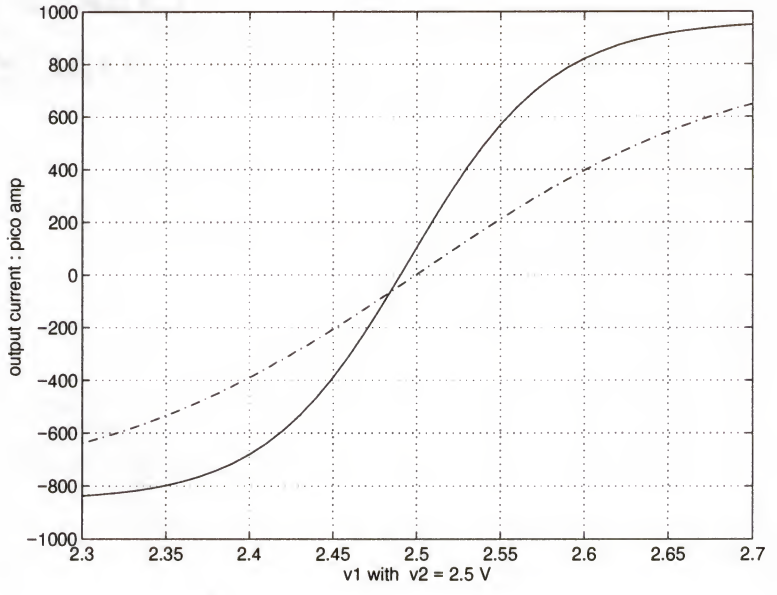


Figure 3.4: The linear range comparisons of transamps with/without extra diodes. solid line : without diodes, dashed line : with diodes.

3.5.2 Wide-range Gilbert Multiplier

A Gilbert multiplier is good candidate to realize the multiplication. Figure 3.5 shows a modified wide-range Gilbert multiplier circuit implementation. With the same technique, diodes are added in order to double the input linear range both for the input signal and its weight value. There is one advantage in choosing a wide-range Gilbert multiplier over a conventional one, where the differential pair input must satisfy [31]:

$$max(Vin, V_{ref}) > min(W_{in}, W_{ref}) \tag{3.35}$$

For a wide-range Gilbert multiplier, there is no constraint between the input voltage (V_{in}) and weight voltage (W_{in}). From Figure 3.5, the output current is

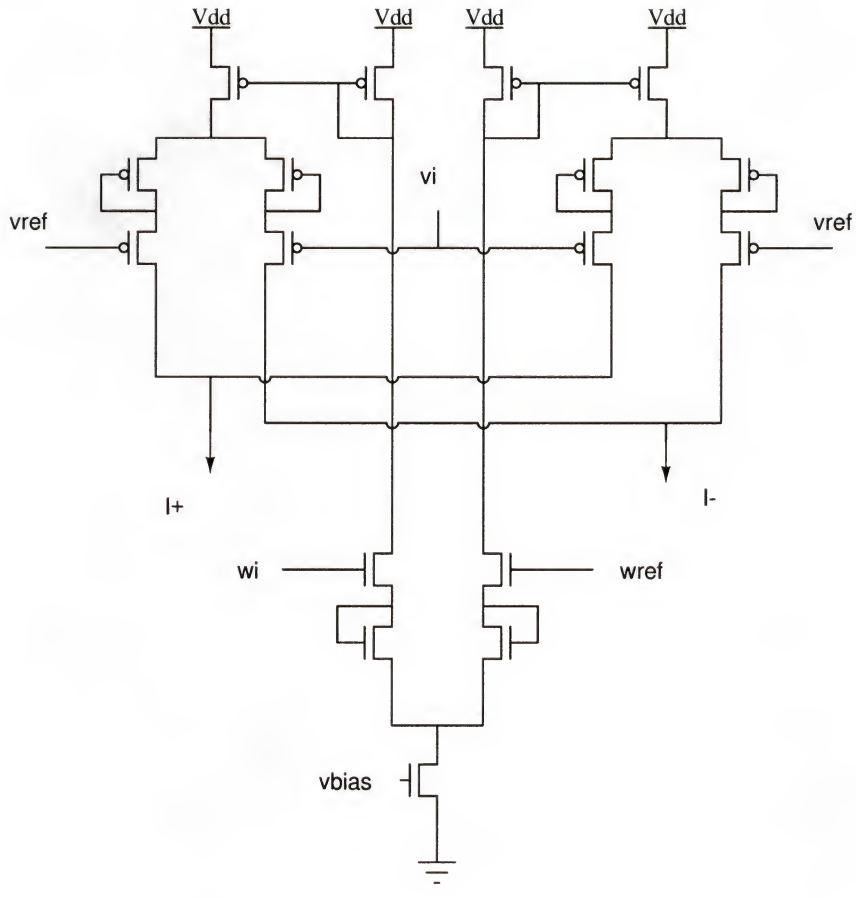


Figure 3.5: Modified wide-range Gilbert multiplier.

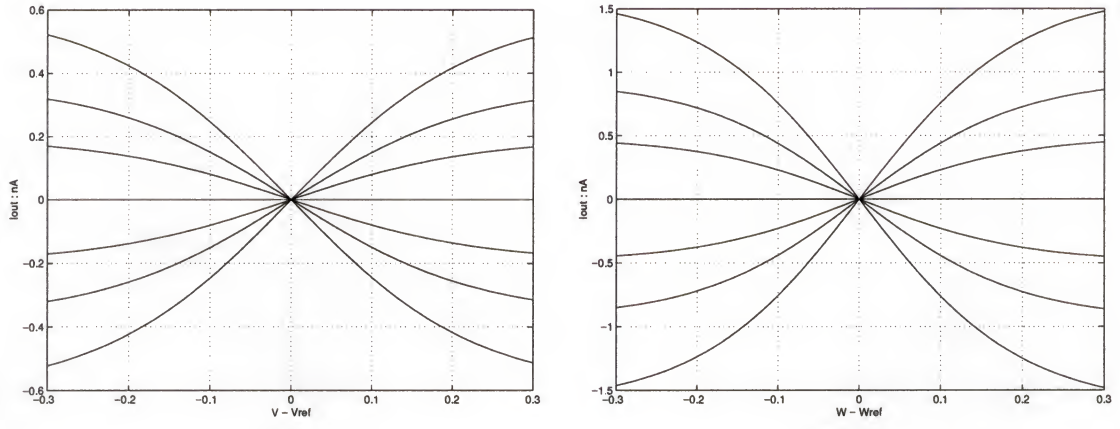
represented as:

$$\begin{aligned} I_{out} &= I_+ - I_- \\ &= I_b \tanh \frac{\kappa(V_{in}-V_{ref})}{4V_T} \tanh \frac{\kappa(W_{in}-W_{ref})}{4V_T} \end{aligned} \tag{3.36}$$

The HSPICE simulation results are shown in Figure 3.6.

As the same as the transamp, when the circuit operates in the sub-threshold region the linear relation between output current and input voltages is guaranteed if and only if these input AC voltages are constrained to 200 mV peak to peak. Note, in the gamma filter, we would like to take the differential currents I_+ and I_- instead of the output current in order to ensure a unidirectional currents of I_+, I_- at all times. These unidirectional currents will be fed to the adaptive circuit to calculate the weight values. For a L-tap gamma memory, the input signal and each individual

tap needs a multiplier which means that L+1 Gilbert multipliers need to be implemented. After that metal lines are simply connected among the output currents to form the positive and negative parts of the output signal, $y^+(t), y^-(t)$.



(a) (b)

Figure 3.6: (a) Output currents as a function of $(w - w_{ref})$ for several values of V-V_{ref}. (b) Output currents as a function of (V-V_{ref}).

3.5.3 Adaptive Weight Circuit

The update rule of the weight in Equation (3.29) depends on the error signal and tap output signal. The output signal, y(t), is represented as a summation of the output currents from L+1 multipliers. Its value can be positive or negative (bidirectional) which will increase the complexity of the circuit design if we didn't separate the output current into two unidirectional currents, I_+ and I_-. Therefore in order to guarantee unidirectional currents we can let

$$y(t) = y^+(t) - y^-(t) \tag{3.37}$$

where both of $y^+(t)$ and $y^-(t)$ are unidirectional currents. With the same procedure, the given desired signal which is a voltage at the input will be represented by two unidirectional currents, $d^+(t)$ and $d^-(t)$ using a linear V-I converter where

$$d(t) = d^+(t) - d^-(t) \tag{3.38}$$

In other words, the desired signal becomes the difference of two unidirectional currents. With the output and desired signals both represented by a pair of unidirectional currents, it becomes straightforward to compute the error signal as the sum of these currents

$$\begin{aligned} e(t) &= d(t) - y(t) = [d^+(t) + y^-(t)] - [e^-(t) + y^+(t)] \\ &= e^+(t) - e^-(t) \end{aligned} \tag{3.39}$$

As soon as the currents, $e^+(t)$, $e^-(t)$, are obtained, we can feed them to the adaptive weight circuit shown in Figure 3.7 where the two bias currents are proportional to the error signals $e^+(t)$ and $e^-(t)$.

Without considering the leaky LMS circuit (negative feedback transamp) on Figure 3.7, the instantaneous weight voltage $w(t)$ is equal to the off-chip capacitor voltage

$$\begin{aligned} C\frac{dw_k(t)}{dt} &= e^+(t)[x_k(t) - x_{ref}] + e^-(t)[x_{ref} - x_k(t)] \\ &= e(t)(x_k(t) - x_{ref}) \end{aligned} \tag{3.40}$$

The adaptive weight equation is valid if the input small signal is within the linear range. The extra negative feedback transamp is added to implement the leaky LMS algorithm. If its bias voltage is set to zero, the circuit reduces to the standard LMS algorithm. By turning on the transconductance of the leaky transamp, the G_{leaky} now is controlled by the bias voltage which also relates to the leaking factor γ. The output voltage of the off-chip capacitor can be rewritten as

$$C\frac{dw_k(t)}{dt} = e(t)(x_k(t) - x_{ref}) - G_{Leaky}w_k(t) \tag{3.41}$$

Comparing Equations (3.29) and (3.41), the learning constant of weight, η_w is inversely related to the value of off-chip capacitance which can be used to control the convergence speed. Meanwhile, the leaky factor γ is related to G_{Leaky} and its value is easy to adjust with a bias voltage to ensure the weight voltage is within the linear range during learning.

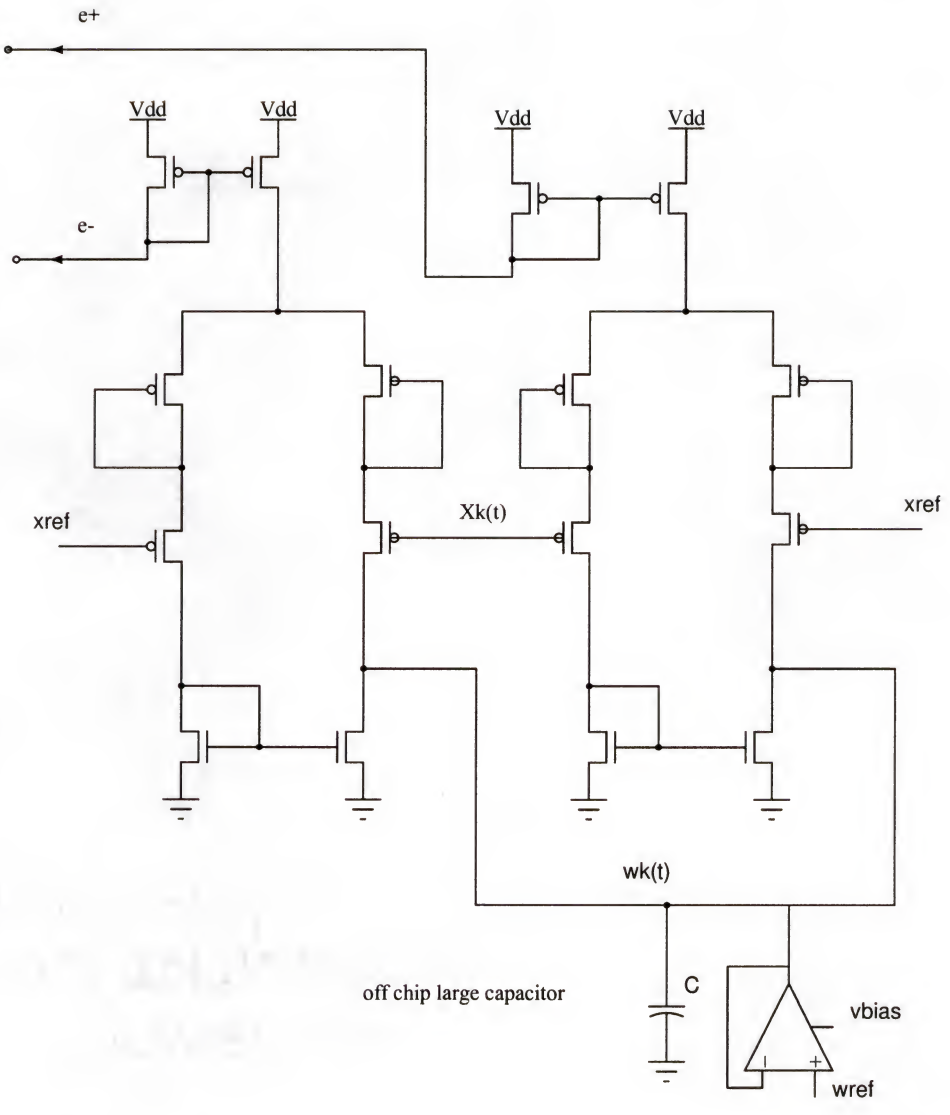


Figure 3.7: The Adaptive weight circuit based on the leaky LMS algorithm.

3.5.4 Absolute-value Circuit

The mean-square-error is usually used to present the performance of the adaptive filter. We use an absolute value circuit for ease of implementation. Figure 3.8 shows a full-wave rectifier. A_1 and A_2 are transamps with the same bias voltage. If V_1 is greater than V_2, then A_1 creates a current flowing into its output. Meanwhile A_2 will shut off the two PMOS current mirror to make the I_2 to equal zero. If V_2 is greater than V_1, then the upper current mirror will be cut off, forcing the I_1 to equal

zero. The output current, I_{out}, then is the sum of I_1 and I_2

$$I_{out} = I_1 + I_2 = I_b tanh(\frac{\kappa|V_1 - V_2|}{4V_T}) \tag{3.42}$$

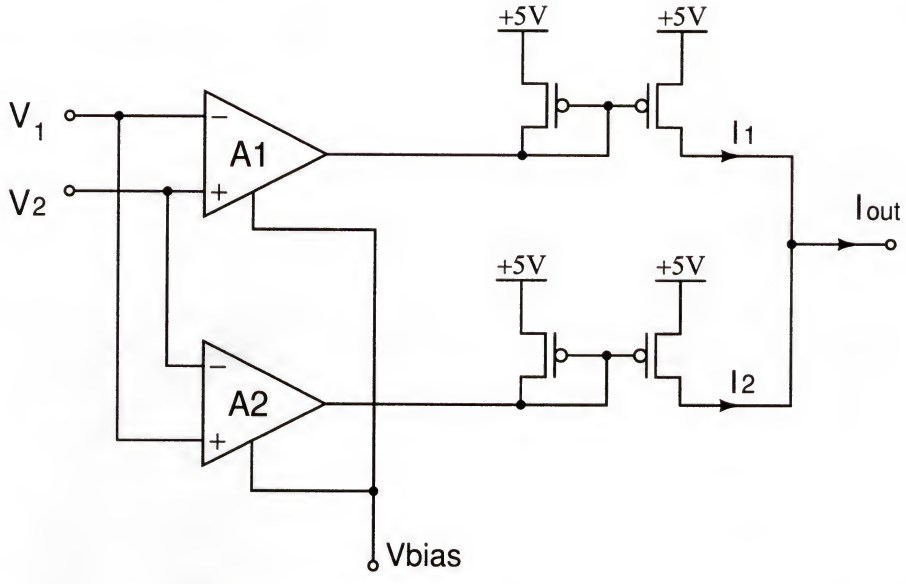


Figure 3.8: Absolute-value circuit.

Each I_1 and I_2 is the half-wave version of input voltage; therefore, I_{out} becomes a full-wave rectifier. The HSPICE simulation result is shown in Figure 3.9

3.5.5 Current Mirror

The DC offset m_e, which is a scalar value, is the error added to error signal $e(t)$. For circuit implementation, a large portion of m_e comes from the mismatch in the current mirror. We use current mirrors to copy identical error signals for the weight update circuits. It has been shown that when current mirrors operate in the sub-threshold region, they provide lower offset than when run above-threshold [48]. On the other hand, circuits operated in the above-threshold region are a better candidate for differential pairs than their below-threshold versions. In order to solve the mismatch problem caused by the Early effect, we increase the channel length of the transistor. Figure 3.10 shows the Early effect measurement both for PMOS and NMOS transistor with L=5 and 25 μm measured from a chip. The Early voltage for

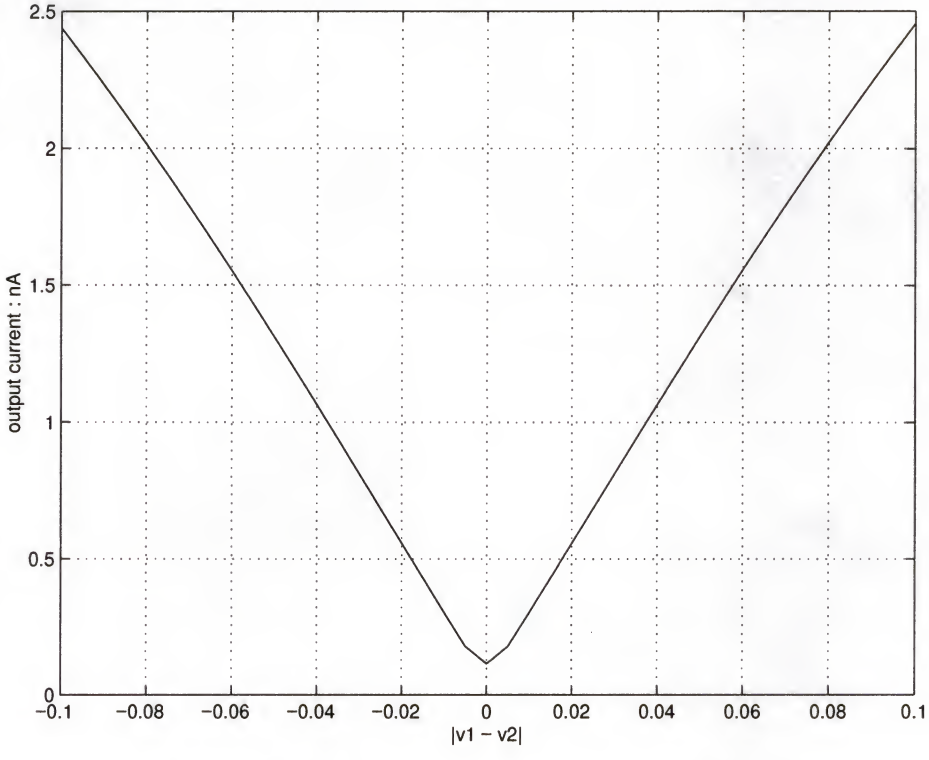


Figure 3.9: HSPICE simulation of absolute-value circuit.

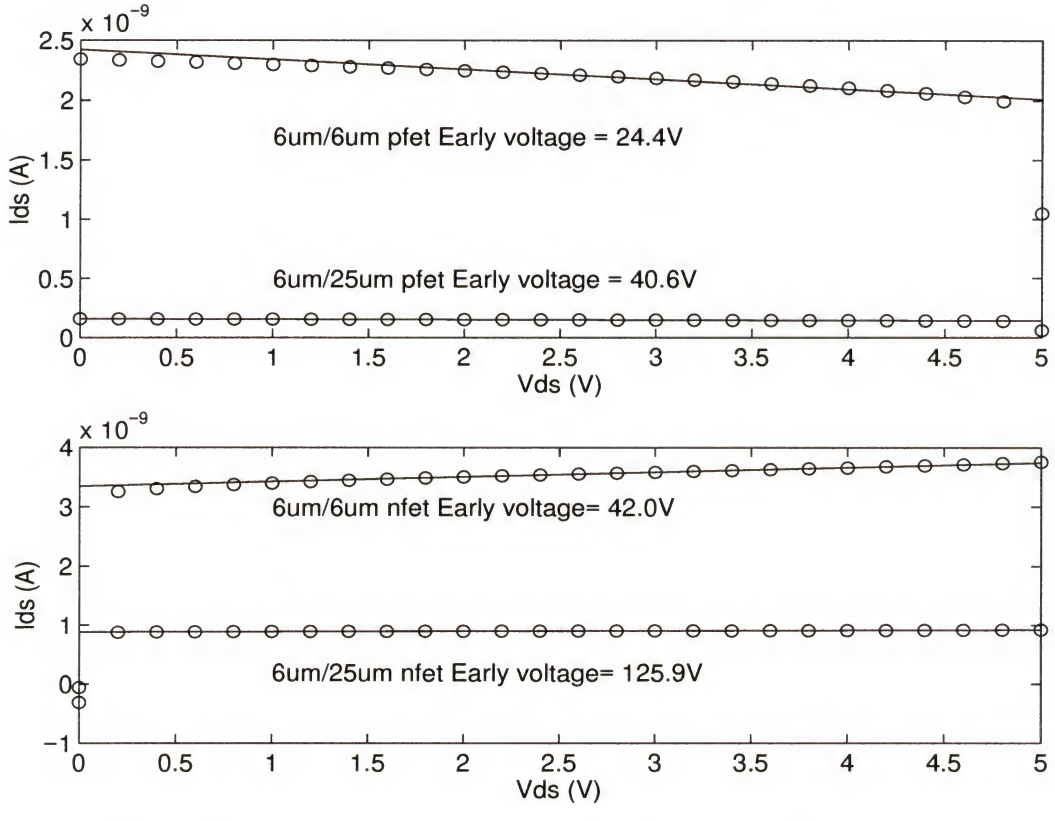


Figure 3.10: The measurements of the early effect for L=5 μm and 25 μm.

NMOS transistors are dramatically improved from 42.0 V to 125.9 V as L increases from 5 to 25 μm.

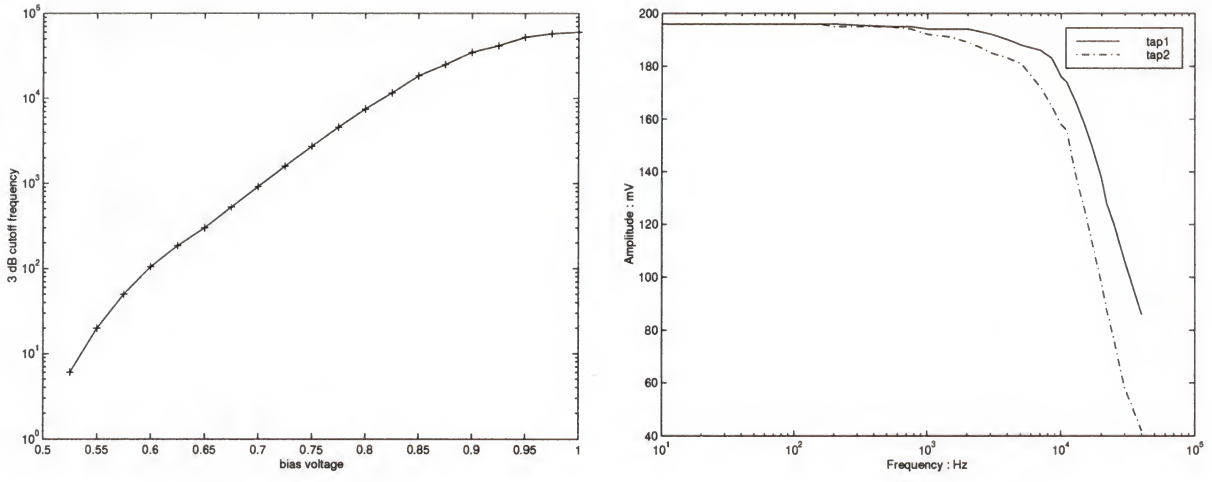
3.6 Chips Test Results

A gamma filter has been fabricated using 2 μm CMOS technology. First, the DC offset, $\mathbf{m_x}$, was measured by inputing a constant DC voltage of 2.5 V. The $\mathbf{m_x}$ value represents the sum of the offsets of each tap. Table 3.2 summarizes the DC offsets $\mathbf{m_x}$ both for the regular transamp and the modified transamp (with extra diodes). These values are the average magnitudes of four chip measurements. It is shown that the modified transamp has a smaller offset. Next, we measured the output current of a single transamp by changing the bias voltage in the sub-threshold region (the threshold voltage is 0.88 V for our measurements). Figure 3.11(a) shows the measured exponential relationship between the output current and the bias voltage in the sub-threshold region. Also the transfer function of the first two gamma taps is shown in Figure 3.11(b). Both transfer functions have the same pole while the second tap has a steep slope of 20 dB/decade. The first 4-tap gamma kernel HSPICE simulation and chip measurement results are shown in Figure 3.12.

In following, we test the gamma chip by choosing two different problems, system identification and noise cancellation problems.

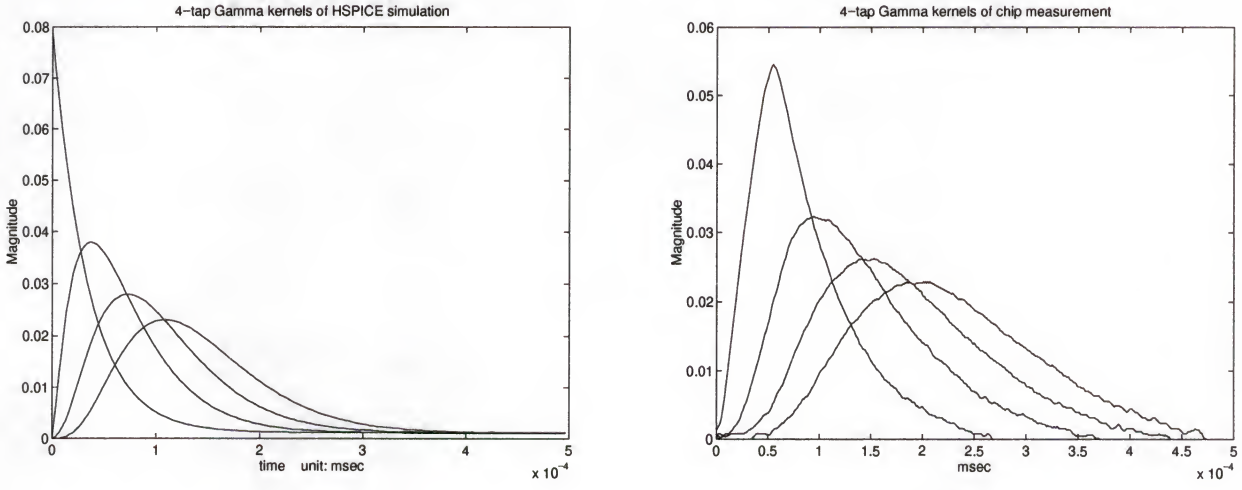
Table 3.2: Measurement of DC offset $\mathbf{m_x}$ unit : mV

transamp	tap 1	tap 2	tap 3	tap 4
without diodes	12.3	19.1	26.4	38.7
with diodes	3.8	5.5	9.8	13.3



(a) (b)

Figure 3.11: Gamma filter measured chip results. (a) Bias voltage vs. 3 dB cutoff frequency. (b) Transfer function of first two taps gamma filter.



(a) (b)

Figure 3.12: (a) Simulated and (b) measured impulse response of gamma filter tap outputs.

3.6.1 System Identification

The system identification includes an unknown plant and an adaptive filter. The goal of this experiment is to adapt the weights of the gamma filter so as to approximate the input-output relation of the unknown plant as closely as possible. The unknown plant is excited by an input source which preferably excites all eigenmodes of the plant. Therefore first we need to generate a broadband white noise signal as an input source. The input signal must contain a proper DC voltage in order to set up the operating point of these circuits properly. Also an AC signal (broadband noise) whose magnitude must be less than 200 mV to ensure these circuits are operated in their linear range. This input signal can be generated using the circuit shown in Figure 3.13. First, a pseudo-random bit stream is generated by a pseudo-random noise chip (MM5437). The typical cycle time of this noise chip is about one minute. The subcircuit1 achieves a flat spectrum noise by filtering the bit stream with a RC circuit. The pseudo-random bit stream and its filtered version are shown in Figure 3.14, Their power spectral density is shown in Figure 3.15.

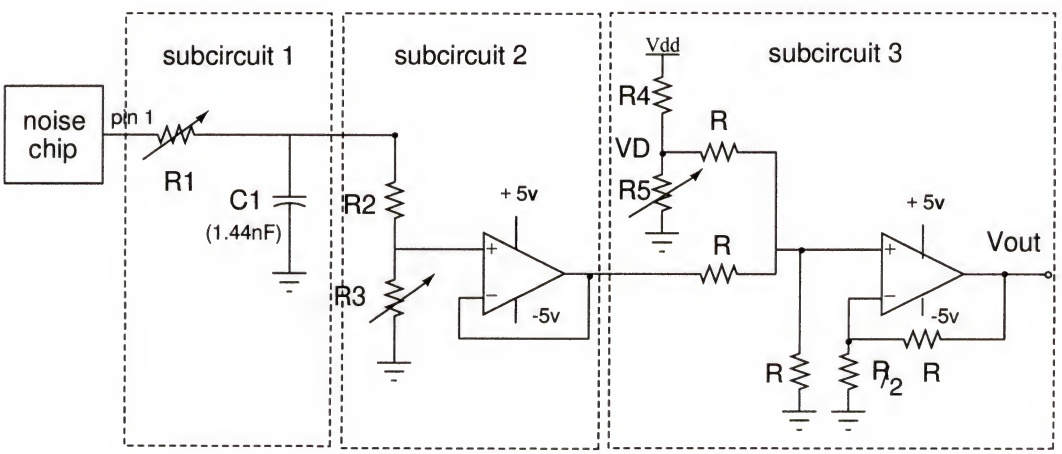


Figure 3.13: Circuit to generate the input noise signal for the system identification problem.

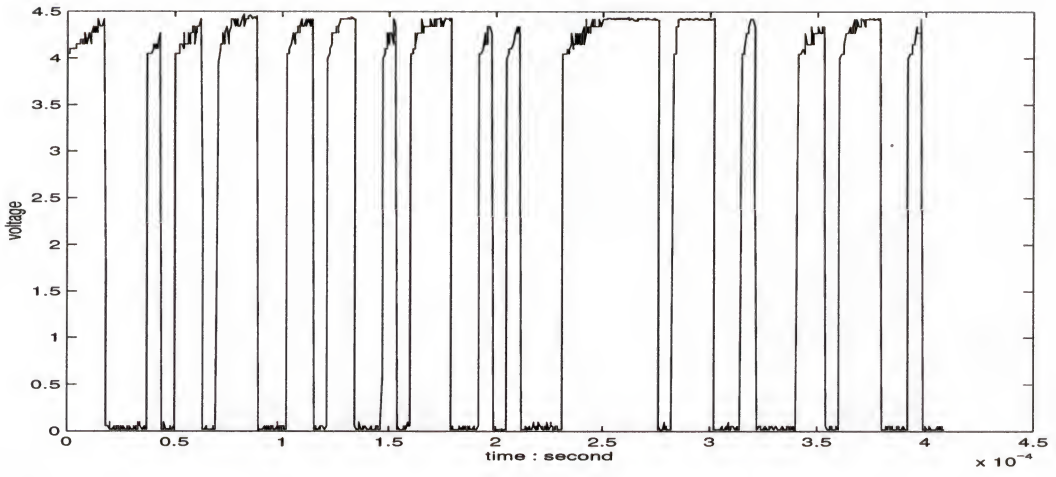


Figure 3.14: Pseudo-random bit stream generated by noise chip.

The subcircuit2 is a voltage divider that ensures the AC signal is less than 200 mV peak to peak. The DC and AC components are added together by subcircuit3.

We choose a unity-gain Sallen-Key low-pass filter shown in Figure 3.16 as the unknown plant [7]. The V_{out} of the Sallen-Key circuit can be represented by V_x at node p as:

$$V_{out} = \frac{V_x}{1 + j\omega RC} \tag{3.43}$$

Further, we can write the KCL at node x as:

$$\frac{V_{in} - V_x}{mR} = \frac{V_x - V_{out}}{R} + \frac{V_x - V_{out}}{1/(j\omega nC)} \tag{3.44}$$

Let $\omega^2 mnR^2C^2 = (f/f_o)^2$, then the transfer function of the Sallen-Key circuit becomes:

$$H(f) = \frac{V_{out}}{V_{in}} = \frac{1}{1 - (f/f_o)^2 + (j/Q)(f/f_o)} \tag{3.45}$$
$$where \quad f_o = \frac{1}{2\pi\sqrt{mn}RC}, \quad Q = \frac{\sqrt{mn}}{m+1}$$

The Sallen-key circuit is a second order filter while the f_o and Q are controlled by the mR and nC values.

In order to test the ability of the gamma filter, a scheme of solving a system

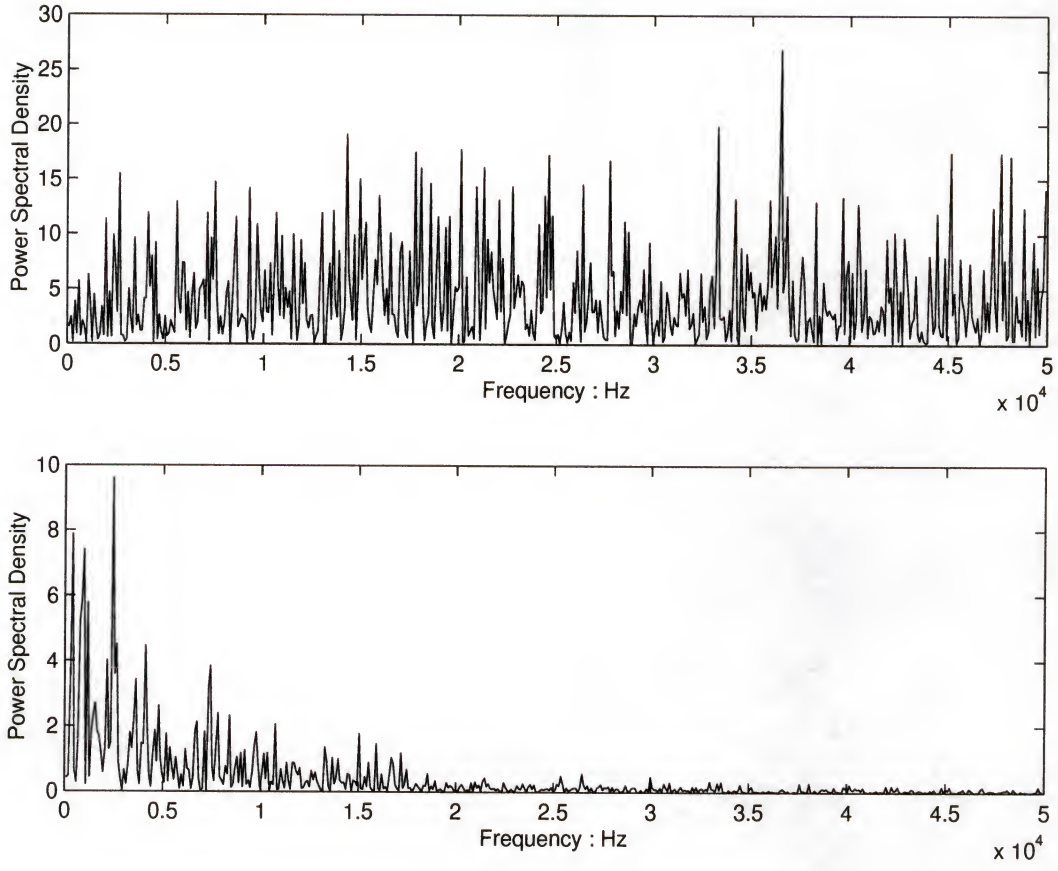


Figure 3.15: Power spectral density of noise chip. Top, before filtering. Bottom, after RC filtering.

identification problem using a 4-tap gamma structure is shown in Figure 3.17. The output of the preprocessor is fed to both the second order Sallen-Key circuit and the 4-tap gamma filter. Now the output of Sallen-Key circuit is used as the desired signal. The error signal is the output of the Sallen-Key circuit subtracted from the output signal generated by gamma filter, then the error signal is fed to update the weights based on the LMS gradient algorithm.

Figure 3.18 shows a sample of the noise input as well as the desired output of the unknown plant. Figure 3.19 shows the output of the filter during and after convergence. The gamma filter is not able to exactly match the Sallen-Key circuit because these two systems have different forms. However, the circuit does a good job in approximating the unknown plant. Figure 3.20(a) shows the changes in the first weight

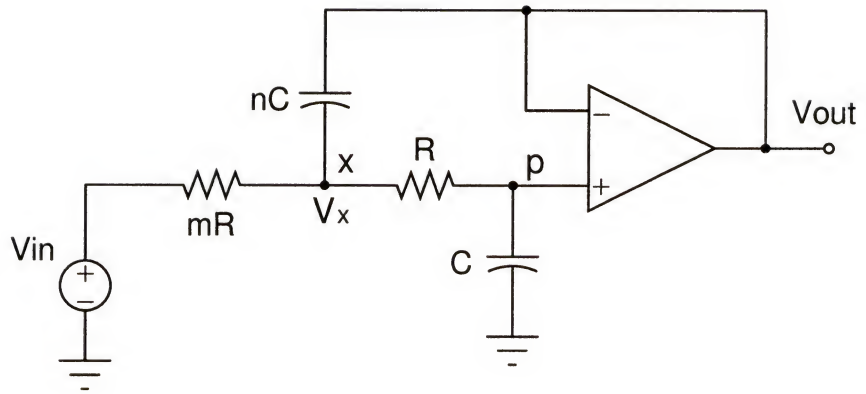


Figure 3.16: Second order Sallen-Key circuit

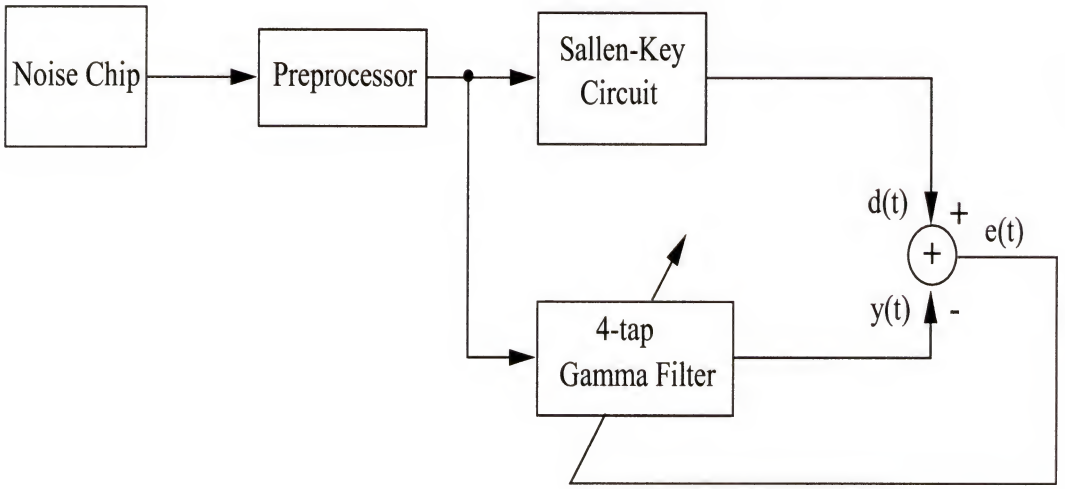


Figure 3.17: System identification flow chart.

during convergence for three different initial values. Figure 3.20(b) shows the path of the weight update for the two weights. As expected for a unimodal energy surface, that same optimal weight values are reached independent of the initial conditions. In order to accurately measure this data, we slowed the learning rate by applying large external capacitors to each weight. With these external capacitors, the convergence rate for this example is on the order of one minute. Much faster convergence can be reached by increasing the learning rate. However, as mentioned in Chapter 2, there is a tradeoff between the learning constant (η_w) and the misadjustment of the gamma filter [49].

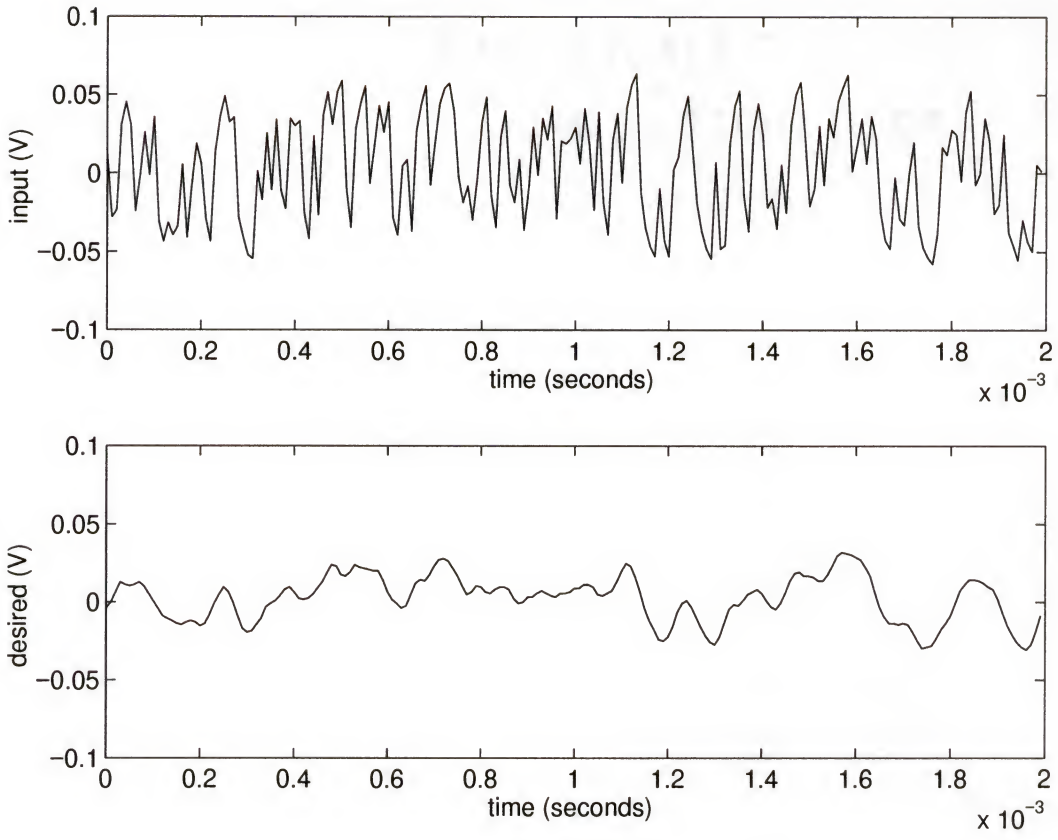
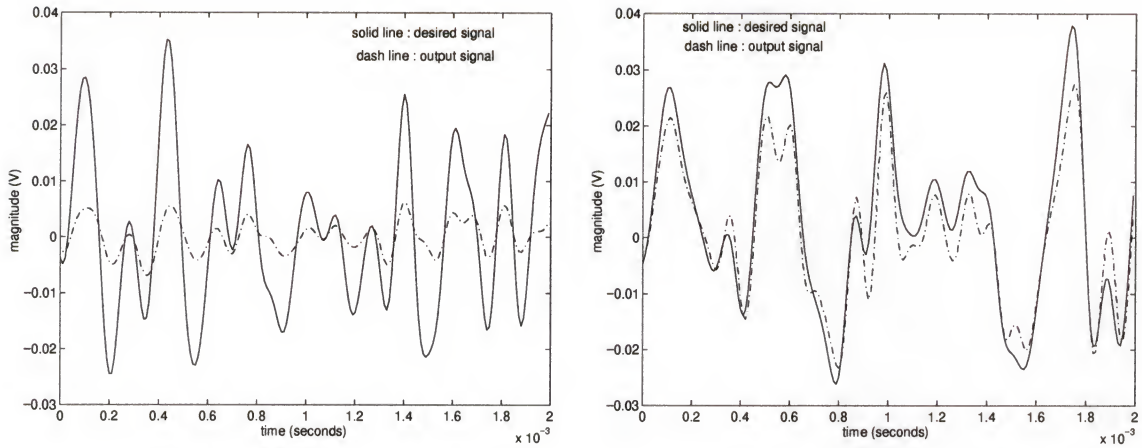


Figure 3.18: The original filtered pseudo-random bit stream is shown at the top. The lower plot shows the the noise after passing through the discrete Sallen-Key circuit. The adaptive filter must mimic the performance of this circuit.

In Figure3.20(b), although the weights become stable, the w_1 has moved outside the linear region (2.4V to 2.6V). Any weight voltage outside the linear range only provides a constant bias current. One way to force these weight voltages to stay within the linear range during the learning is to enable the leaky LMS algorithm by turning on the feedback transamp of the adaptive circuit. Figure 3.21 shows the weight tracks of leaky LMS with off-chip capacitances equaling 0.47 μF. Finally, an absolute value circuit is used to produce the instantaneous absolute error instead of the instantaneous mean-square-error. Figure 3.22 shows the instantaneous absolute error decreasing during the adaptation.

From these test results, we conclude the following:



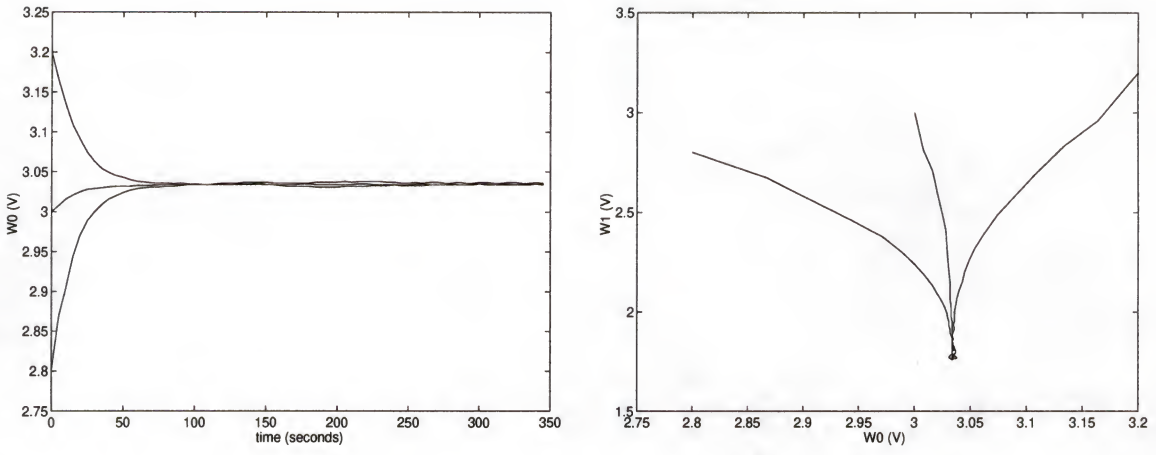
(a) (b)

Figure 3.19: (a) A sample of the desired and the output signal before convergence. (b) Another sample of the same signals after convergence.

1. Because searching the non-convex performance surface for τ is difficult, the τ value is determined by hand with an off-chip voltage bias. Therefore during the learning, we need to choose a reasonable τ by hand based on the monitored the output and the desired signals displayed on the scope. In other words, a global search of τ is done by hand. Experience shows that even with different τ values, the gamma filter still converges to a good solution.

2. The final weights based on the leaky LMS algorithm with a off-chip capacitance equaling 1.2 pF are w_0=2.52V, w_1=2.61V, w_2=2.56V, w_3 =2.52V. All of the weights converge to be within the linear range (from 2.4 V to 2.6 V).

3. Figure 3.21 shows that each individual weight track is not the same even they have identical learning rate (off-chip capacitors). This is because of the eigenvalue spread of the auto-correlation matrix among the taps. If we look at an individual weight, the convergence speed is not the same for different initial value. For instant, the w_1 converges after six minutes for initial weight equaling 3 V, the w_1 converges after two minutes for initial weight value equaling 2



(a) (b)

Figure 3.20: (a) The trajectory of the weight w_0 during convergence for three different initial values. (b) The trajectory for both weights w_0 and w_1.

V. However, the learning speed is less important. Making sure the weights converge to their optimal value is more important for the typical audio frequency applications we are considering.

4. All of the leaky factors,γ, are set to the same constant. For best results, the γ of each individual weight can be chosen differently.

5. There are several bias voltages which need to be set properly. One of them is the bias voltage of the modified Gilbert wide-range multiplier. Within the linear range, the relationship between multiplier bias voltage and weight voltage is linear. Doubling the bias voltage usually reduces the weight voltage by half.

3.6.2 Noise Cancellation

Adaptive filters can be used in interference canceling problems. Some of the earliest work was performed by Howells in 1960 [49]. The basic concept of the noise cancellation is shown in Figure 3.23. The primary source represents a signal $s(t)$ transmitted over a channel to a sensor that receives the signal and an uncorrelated noise $n_o(t)$, The goal is to recover the signal using adaptive gamma filter to cancel the unwanted noise $n_o(t)$. In order to achieve the goal, we need another noise source

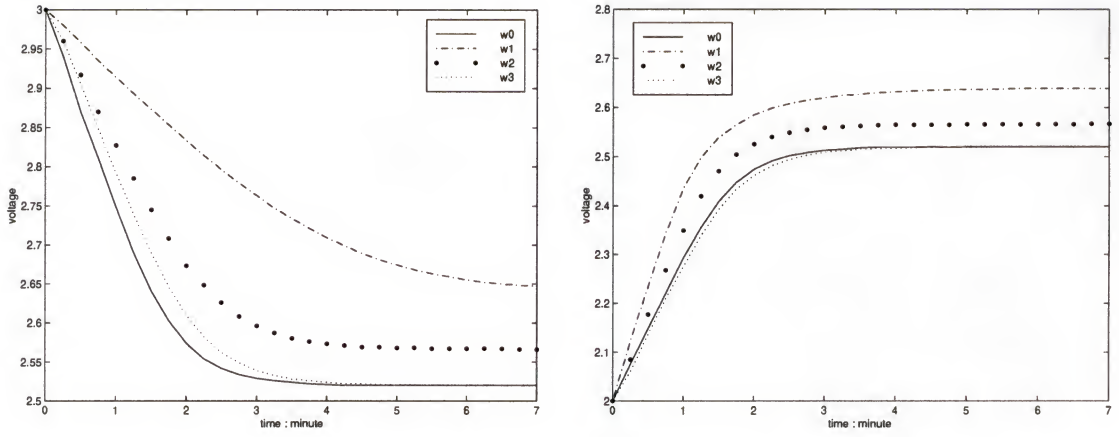


Figure 3.21: Weight tracks based on the leaky LMS algorithm. (a) Initial weight voltages equal 3 V. (b) Initial weight voltages equal 2 V.

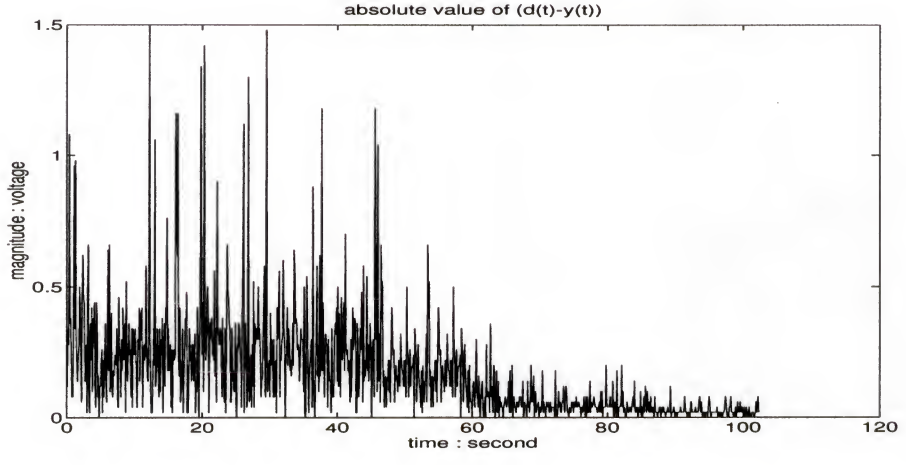


Figure 3.22: The absolute instantaneous error signal for the system ID problem.

called $n_1(t)$ as the reference input. We assume $n_1(t)$ is uncorrelated with the signal $s(t)$ but correlated in some unknown but linear way with the noise $n_o(t)$. The noise $n_1(t)$ then is fed into the adaptive gamma filter to produce an output signal $y(t)$, that is a close replica of $n_o(t)$ on the primary input. The system output $\varepsilon(t)$ then is equal to the the primary signal minus the output signal

$$\varepsilon(t) = s(t) + n_o(t) - y(t) = s(t) + n_o(t) - \sum_{k=0}^{L} w_k \cdot n_{o,k}(t) \tag{3.46}$$

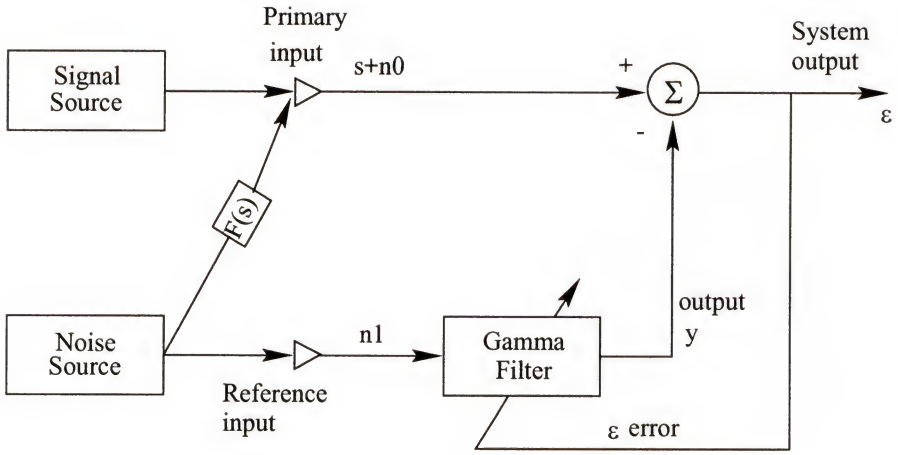


Figure 3.23: Noise cancellation chart.

Unlike the system identification problem, for noise cancellation system the practical objective is to produce a system output, $s(t) + n_o(t) - y(t)$, that is a best fit in the least square sense to the signal $s(t)$. Assume $s(t)$, $n_o(t)$, $n_1(t)$, and $y(t)$ are statistically stationary and have zero means. Squaring Equation (3.46) we get

$$\varepsilon(t)^2 = s(t)^2 + 2s(t)(n_o - y(t)) + (n_o - y(t))^2 \tag{3.47}$$

Taking the expected value of both sides:

$$E[\varepsilon(t)^2] = E[s(t)^2] + 2E[s(t)(n_o - y(t))] + E[(n_o - y(t))^2] \tag{3.48}$$

According to our assumptions, the signal $s(t)$ is uncorrelated with $n_o(t) - y(t)$, the second term can be dropped, then we get

$$E[\varepsilon(t)^2] = E[s(t)^2] + E[(n_o - y(t))^2] \tag{3.49}$$

The power of the signal $s(t)$ will not be affected as the filter is adjusted to minimize $E[\varepsilon(t)^2]$. When the adaptive gamma filter is adjusted so that $E[\varepsilon(t)^2]$ is minimized, $E[(n_o(t) - y(t)^2]$ is therefore also minimized. The output of the system $\varepsilon(t)$ is then a best least square estimate of the signal $s(t)$.

In order to test the gamma chip, we set up a simple example as follows:

$$\begin{aligned} s(t) &= A\sin(\omega_o t) \\ n_o(t) &= B\sin(\omega_1 t) \\ n_1(t) &= C\sin(\omega_1 t + \theta) \end{aligned} \tag{3.50}$$

Where ω_o is the signal frequency and ω_1 is another frequency which represents the channel noise. The signals $s(t)$, $n_o(t)$, and $n_1(t)$ satisfy the assumptions we stated above. Figure 3.24 shows the primary signal and reference signal. The output of the gamma filter and the primary input are connected to an off-chip op-amp subtractor to obtain the system output signal $\varepsilon(t)$. The optimal time constant, τ, of the gamma filter can be determined by monitoring the signal $s(t)$ and system output $\varepsilon(t)$ on the scope at the same time. Figure 3.25 shows the result after the gamma filter converged.

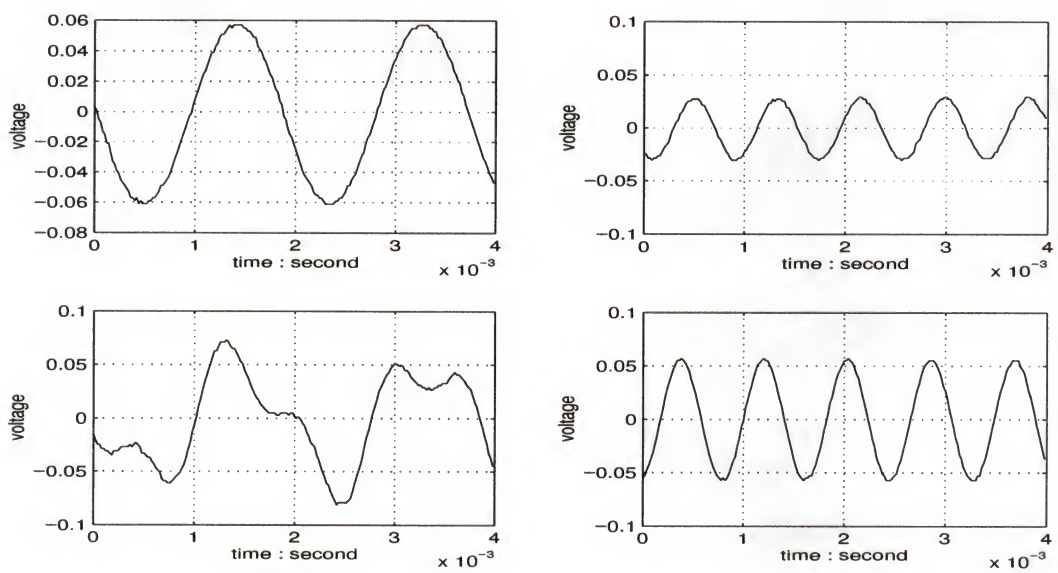


Figure 3.24: Top left: signal $s(t)$. Top right: $n_o(t)$. Bottom left: $s(t)+n_o(t)$. Bottom right: $n_1(t)$.

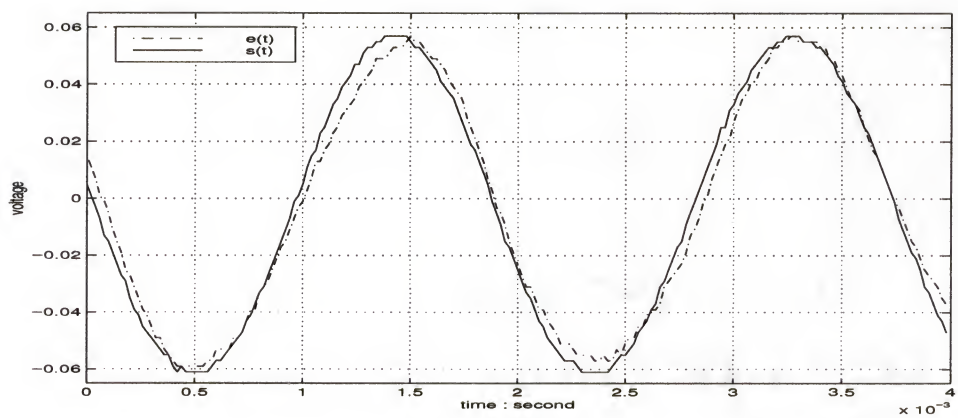


Figure 3.25: The signal source, $s(t)$, and the system output signal $\varepsilon(t)$ after convergence.

CHAPTER 4
CONTINUOUS-TIME LAGUERRE FILTER

We discuss another continuous-time generalized feedforward filter - the Laguerre filter in this chapter. The gamma kernels we mentioned in the previous chapter are linearly independent of one other [1] but do not form an orthogonal set. There are several popular orthogonal kernels including the Laguerre, Legendre, Jacobi, and Kautz [14, 25]. We are more interested in the Laguerre kernels than the other orthogonal kernels because of their relative simplicity. Actually, the Laguerre filter can be viewed as one real pole (one time constant) system like the gamma filter. On the other hand, the Legendre, Jacobi, and Kautz can have more than one pole (real or complex). First we review the Laguerre kernels in the time and frequency domains. Unlike the gamma kernels which are always positive values, the Laguerre kernels fluctuate with positive and negative values. By cascading all-pass filters with a low-pass filter in front, we can construct the Laguerre memory structure. Later, we show the circuit design of the Laguerre filter with active transamps. We simulate and test the performance of the Laguerre filter for the system identification problem. We conclude that the Laguerre filter has the advantages of faster convergence speed, an orthogonal basis set, and higher signal to noise ratio (S/N) than the gamma. However, the Laguerre suffers from larger DC offsets, and requires more silicon area

[1] For any individual gamma kernel, $g_k(t)$, can not be represented as the linear combination of the other gamma kernels. It turns out, with arbitrary coefficients a_m

$$g_k(t) \neq \sum_{m=0, m\neq k}^{\infty} a_m \cdot g_m(t)$$

than the gamma.

4.1 The Laguerre kernel and Laguerre filter

The Laguerre kernels, which are complete in the Hilbert space and have a novel orthonormal characteristic, can be obtained by orthonormalization of the sequence

$$(t/\tau)^k e^{-t/\tau} \qquad 0 \leq t < \infty \tag{4.1}$$

whose first few kernels and general expressions are

$$\begin{aligned} l_0(t) &= \sqrt{2/\tau}e^{-t/\tau} \\ l_1(t) &= \sqrt{2/\tau}[1-2(t/\tau)]e^{-t/\tau} \\ l_2(t) &= \sqrt{2/\tau}[1-4(t/\tau)+2(t/\tau)^2)]e^{-t/\tau} \\ l_k(t) &= \sqrt{2/\tau}e^{t/\tau}\frac{d^k}{dt^k}[\frac{(-t)^k}{n!}e^{-2t/\tau}] \end{aligned} \tag{4.2}$$

where τ is the time constant. According to the orthonormal property we can prove

$$\int_0^\infty l_m(t)\cdot l_n(t)dt = \delta_{mn} = \begin{cases} 1 & m=n \\ 0 & m\neq n \end{cases} \tag{4.3}$$

In the time-domain, the Laguerre kernel described in Equation (4.2) is a summation of k+1 terms with a different time dimension multiplied by a common exponential delay term in front. It is not easy to interpret the formula at first glance. Figure 4.1 shows the first four Laguerre kernels, the first Laguerre kernel, k=0, is an exponential decay function which has a non-negative value. With $k \geq 1$ the Laguerre kernels not only fluctuate with positive and negative values but also dampen out with time in a manner similar to Bessel functions. The Laguerre kernels have more than one peak. In fact, the number of minimum/maximum of Laguerre kernels is proportional to the order L. Also the min/max alternate along the time axis.

In the s-domain, the Laguerre kernels are given by:

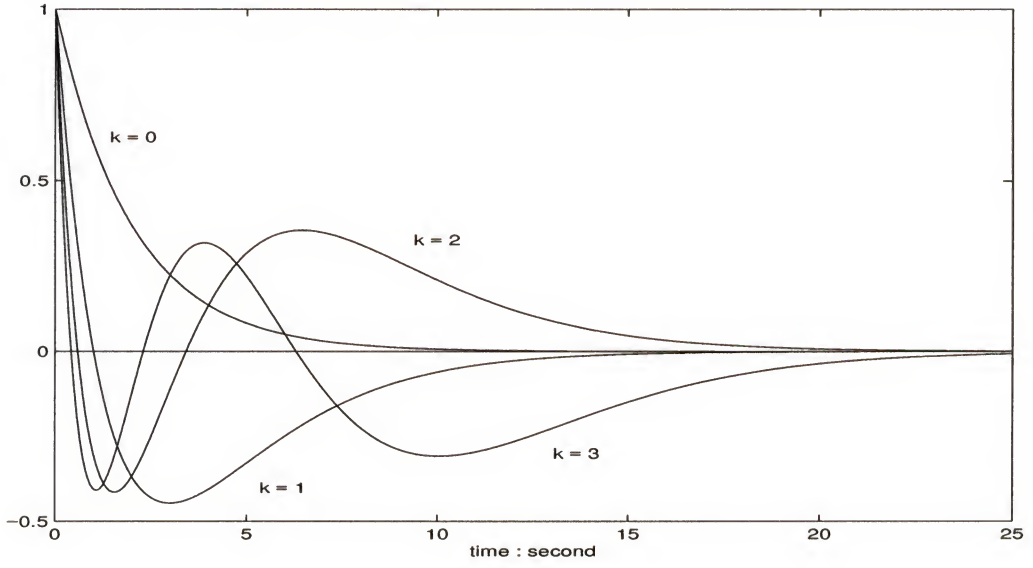


Figure 4.1: Continuous-time Laguerre kernels.

$$L_k(s) = \sqrt{2\tau} \cdot \frac{(\tau s - 1)^k}{(\tau s + 1)^{k+1}} \tag{4.4}$$

Note, the first tap of Laguerre kernels start with $k = 0$, which is a first-order low-pass filter. For $k \geq 1$, the transfer function of the Laguerre kernels becomes an all-pass filter. The Laguerre memory structure is shown in Figure 4.2, which is composed of a first-order low-pass filter followed by cascaded first-order all-pass filters with the same pole value $1/\tau$.

There are two advantages of a Laguerre memory over the gamma. First,

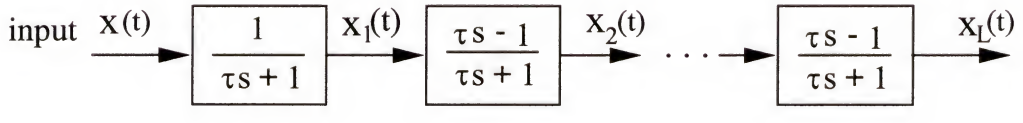


Figure 4.2: Continuous-time Laguerre memory structure.

when the power spectrum of the input signal is constant, then the outputs of the Laguerre memory are orthogonal to one other. Second, because the Laguerre kernels

are all-pass filters (except for the first tap), the amplitude of the signal will not be attenuated while propagating along the structure. This characteristic is important in analog VLSI implementations where noise in the computation must be considered since noise sensitivity increases as the signal amplitude decreases.

The weight update rule of the Laguerre filter using the LMS algorithm is identical to that of the gamma filter, while the τ update using the steepest descent method becomes much more complicated:

$$\frac{d\mu(t)}{dt} = \frac{\eta_\mu}{\mu(t)} \sum_{k=0}^{L} w_k(t)[x_k(t) - (k-1)x_{k-1}(t) * g(t) - kx_k(t) * g(t)] \tag{4.5}$$

where g(t) is the impulse response of the gamma unit, and '*' denotes the convolution operator.

The Wiener-Hopf equations for the Laguerre filter is the same as the gamma filter shown in Chapter 3. Like the gamma filter, we can derive the state-space Laguerre filter as follows. For k = 0, Equation (4.4) becomes

$$\begin{aligned} sL_0(s) &= \frac{1}{\tau}[\sqrt{2\tau}U(s) - L_0(s)] \\ \frac{dl_0(t)}{dt} &= \frac{-1}{\tau}l_0(t) + \sqrt{2/\tau}U(t) \end{aligned} \tag{4.6}$$

where $U(t)$ is the input signal, for $k \geq 1$

$$sL_k(s) = -\frac{1}{\tau}L_k(s) + sL_{k-1}(s) - \frac{1}{\tau}L_{k-1}(s) \tag{4.7}$$

Equation (4.7) can be represented by

$$\frac{dl_k(t)}{dt} = -\frac{1}{\tau}l_k(t) - \frac{2}{\tau}[l_{k-1}(t) + l_{k-2}(t) + ... + l_0(t)] + \sqrt{2/\tau}U(t) \tag{4.8}$$

Combining Equation (4.7) and (4.8),then the overall state-space equations become

$$\frac{d}{dt}\begin{bmatrix} l_0(t) \\ \cdot \\ \cdot \\ l_k(t) \end{bmatrix} = \begin{bmatrix} -1/\tau & 0 & 0 & 0 \\ -2/\tau & -1/\tau & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot \\ -2/\tau & \cdot & -2/\tau & -1/\tau \end{bmatrix} \begin{bmatrix} l_0(t) \\ \cdot \\ \cdot \\ l_k(t) \end{bmatrix} + \begin{bmatrix} \sqrt{2/\tau} \\ \cdot \\ \cdot \\ \sqrt{2/\tau} \end{bmatrix} U(t) \tag{4.9}$$

The output signal equation is

$$y(t) = \begin{bmatrix} w_0 & \cdot & \cdot & w_k \end{bmatrix} \begin{bmatrix} l_0 \\ \cdot \\ \cdot \\ l_k \end{bmatrix} \tag{4.10}$$

Note, for the Laguerre filter, the output signal does not include the input signal, $U(t)$.

4.2 Circuit Implementation of the Laguerre Filter

The circuits needed in the Laguerre filter are the same as the gamma filter except now we need to implement a Laguerre memory structure instead of the gamma structure. Unlike the gamma whose transfer function, $H_k(s)$ is identical, the Laguerre structure needs a low-pass filter in the front cascaded by all-pass filters. The pole of these filter is the same. Let's start with the circuit shown in Figure 4.3. Using Kirchoff's current law at node V_{low}:

$$V_{low}sC + \frac{(V_{low} - V_{in})}{R} = 0 \tag{4.11}$$

the transfer function between V_{low} and V_{in} is

$$\frac{V_{low}}{V_{in}} = \frac{1}{\tau s + 1} \qquad \tau = RC \tag{4.12}$$

If we apply the Kirchoff's current law at node V_{out}, it becomes:

$$\frac{V_{low} - V_{out}}{R1} + \frac{V_{low} - V_{in}}{R1} = 0 \tag{4.13}$$

After rearrangement, the transfer function between V_{out} and V_{in} is given by:

$$\frac{V_{out}}{V_{in}} = \frac{\tau s - 1}{\tau s + 1} \tag{4.14}$$

Taking outputs at nodes V_{out} and V_{low}, we guarantee the pole of the low-pass unit has the same value of the all-pass unit in any situation. The passive resistors

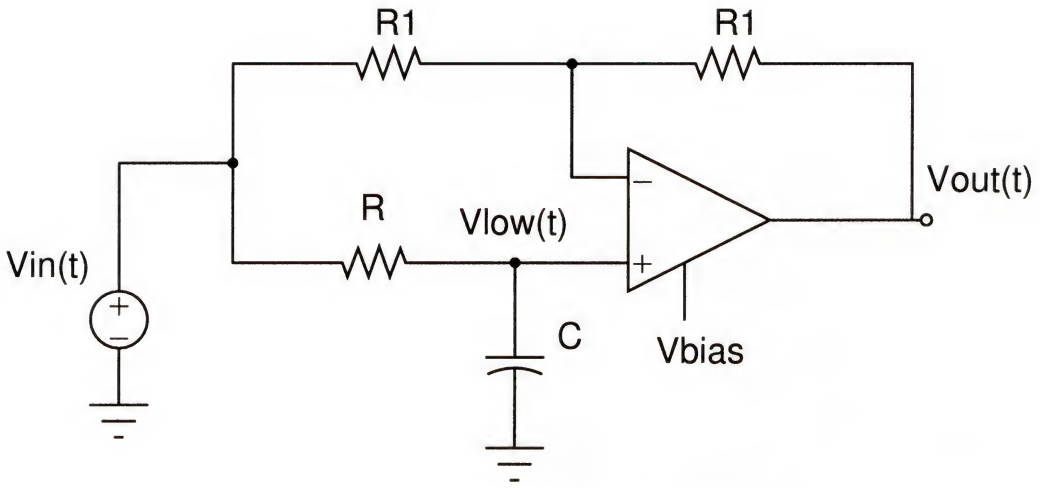


Figure 4.3: First-order all-pass circuit.

later can be replaced by active element resistors (transamps with negative feedback). This replacement will not only avoid large silicon area on chip but also can provide an adaptive transconductance which allows us to control the pole location during learning. Figure 4.4 shows the final circuit for the Laguerre unit. Although using a transamp to replace passive resistors saves chip area and provides an adaptive transconductance, the price paid is a worse mismatch between these two transamps which will cause an non-ideal all-pass circuit in term of a more serious DC offset $\mathbf{m_x}$ problem. In Section 4.3, we will show the offset both from an HSPICE simulation and from a chip measurement.

4.3 Simulations and Test Results

A 4-tap Laguerre filter chip was fabricated in 2μm CMOS technology. First we address the offset $\mathbf{m_x}$ both in HSPICE simulation and in a chip measurement, and summarize them in Table 4.1. The HSPICE simulation can only show the systematic components of $\mathbf{m_x}$ but the actual chip measurements show a much worse value of $\mathbf{m_x}$ due to the random component. Again the $\mathbf{m_x}$ data is based on the average of four chip measurements. The most serious problem for a large offset is that it will drive the circuit out of the linear range when learning begins. Therefore, the adaptive filter

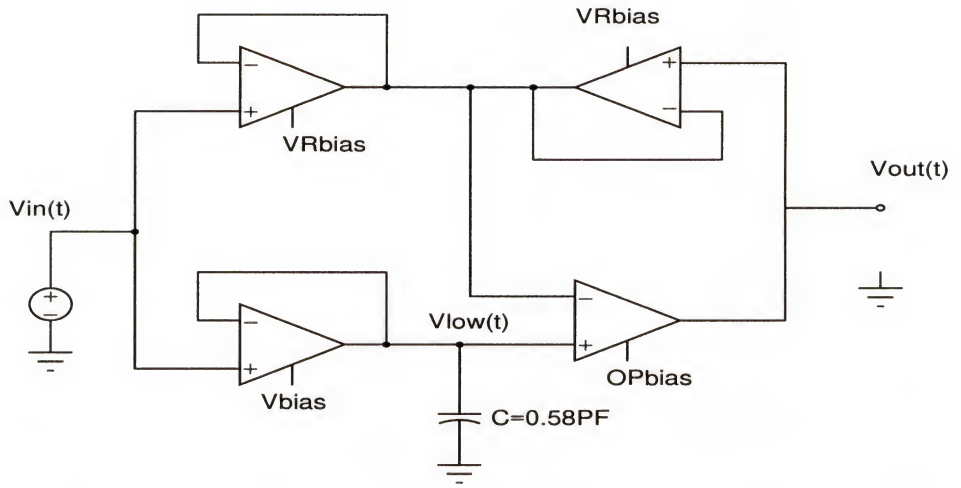


Figure 4.4: First-order all-pass circuit with active transistors.

Table 4.1: Laguerre filter DC offset $\mathbf{m_x}$ unit : mV

method	tap 1	tap 2	tap 3	tap 4
HSPICE	6	16	27	36
chip test	6	28	41	50

has no chance to recover. Comparing $\mathbf{m_x}$ between the gamma and the Laguerre, the large offset arises from the mismatch between the transamps acting as resistors.

The transfer function of a 4-tap Laguerre filter of HSPICE simulation and chip measurements are shown in Figure 4.5. Note that in order to show the characteristic of an all-pass filter, the input signal is sent directly to the all-pass filter, the low-pass curve is measured at node V_{low}.

The measurement result shows that the Laguerre filter is an all-pass filter with frequency below 25 KHz. Figure 4.6 shows that within the passband of the front low-pass filter an input sine wave is ideal delayed but not attenuated by a 4-tap Laguerre filter.

In order to compare the gamma and the Laguerre filters, we simulated the continuous-time implementation of both the gamma and the Laguerre filters for a system identification problem. White noise is filtered by an unknown plant and an

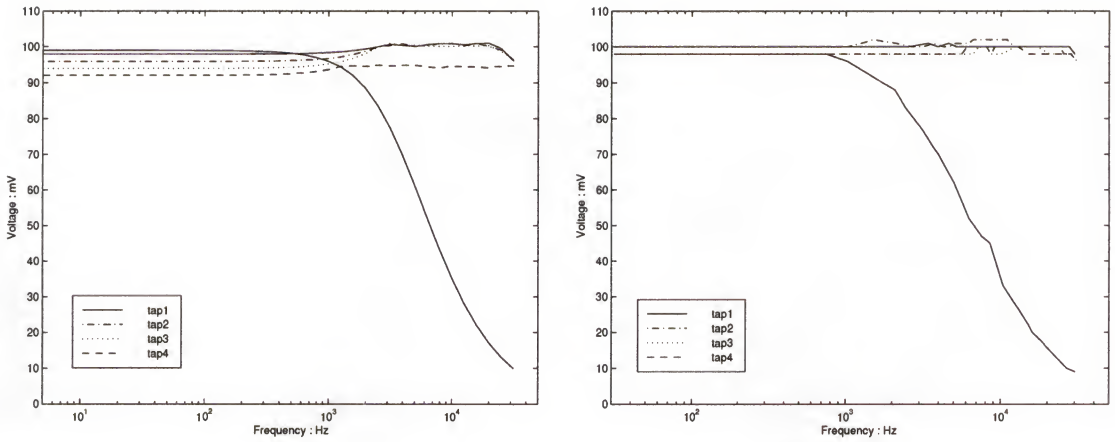


Figure 4.5: The 4-tap Laguerre filter transfer function. (a) HSPICE simulation result. (b) Chip measurement.

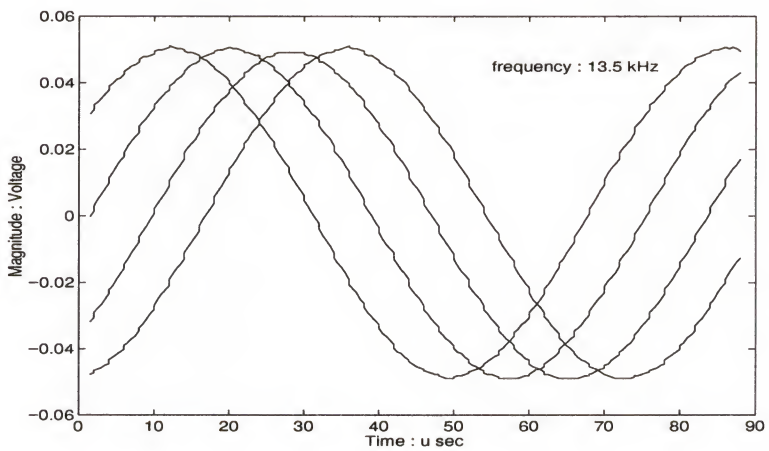


Figure 4.6: Measured results of 4-tap Laguerre filter.

adaptive system is set up as shown in Figure 4.7 to identify the unknown transfer function.

We show in Figure 4.8, a comparison of the convergence rates for the system using the two different filters. As expected, the Laguerre shows a much faster convergence rate. However, in an analog implementation, we are not as much concerned with convergence speed as we are with the effects of noise and offsets in the system. We have added random noise with a certain standard deviation to the output of each tap to simulate the effects of real-world noise and random parameter variations.

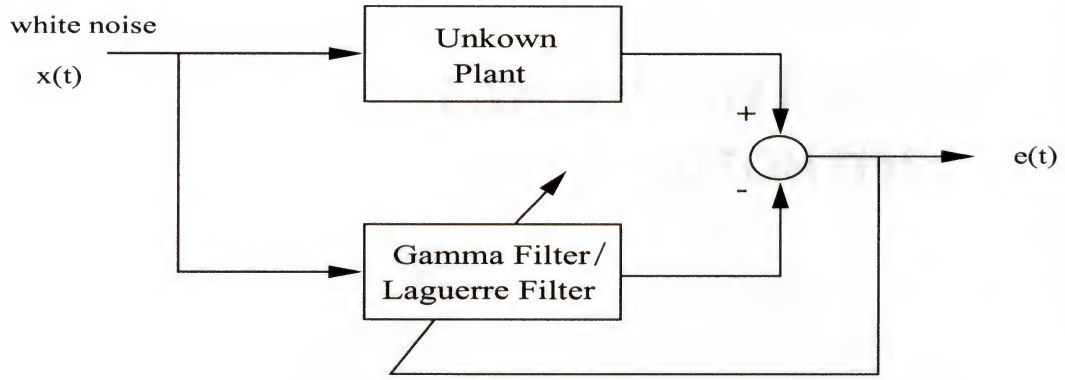


Figure 4.7: Block diagram for system identification simulation.

The results shown in Figure 4.9 indicate that the Laguerre is more robust than the gamma for the same amount of noise, making it a much better candidate for analog implementation.

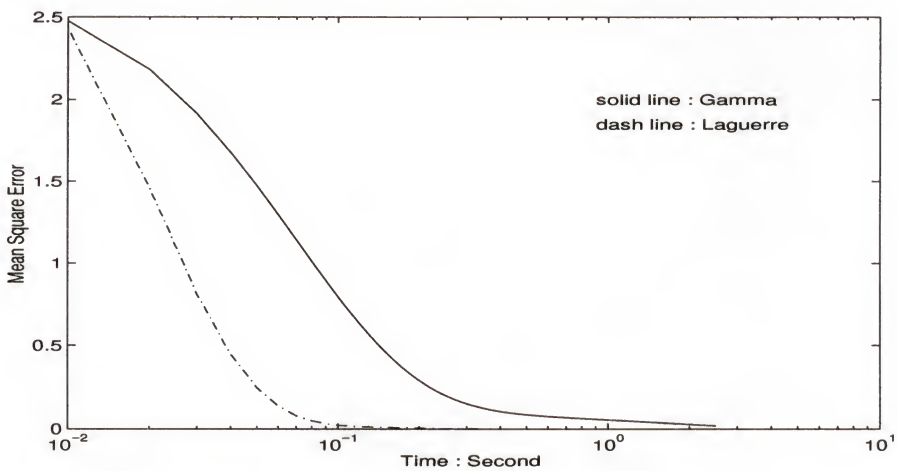


Figure 4.8: The faster learning rate of the Laguerre vs. the gamma.

We test the adaptive 4-tap Laguerre filter with the same system ID problem shown for the gamma filter. We initialize all weights to 2.0 V or 3.0 V at t=0. The final steady state voltages for weights are, w_0=2.522V, w_1=2.466V, w_2=2.549V, w_3=2.520V with DC input signal bias at 2.5V. These voltages are all within the

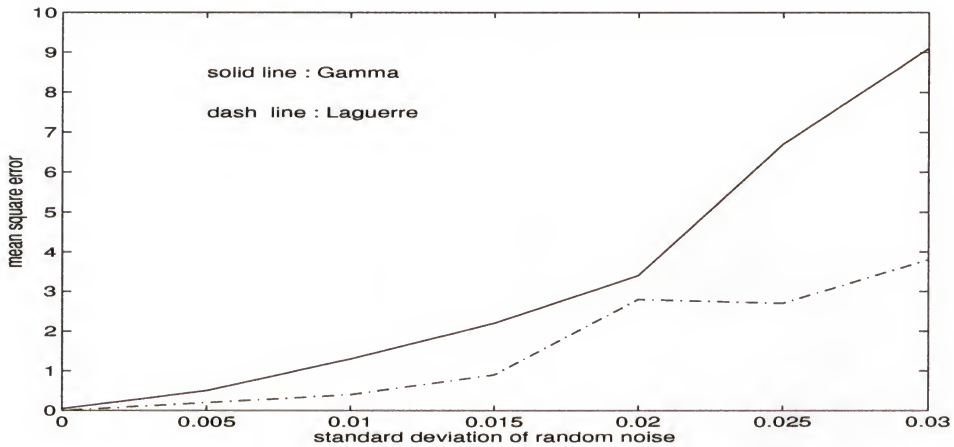


Figure 4.9: The comparison of MSE vs. the amount of added random noise for the Laguerre and gamma. The Laguerre shows a lower MSE for the same amount of added noise at the tap outputs.

linear range. These weight tracks are shown in Figure 4.10. By observing each individual weight, we find out their convergence speed is almost the same. We explain this situation in two ways. First, from Chapter 2, we know that the convergence speed of the weight depends on its corresponding eigenvalue. Since the Laguerre kernels are orthogonal, it might have low eigenvalues spread. Therefore the weight convergence speed is almost the same for every tap. Second, from the point of view of circuit design, we know the convergence speed is proportional to the magnitude of tap output. Basically the Laguerre filter is an all-pass filter. It turns out the amplitudes of Fourier components at each tap are identical, and the convergence speed roughly the same. On the contrary, the gamma filter is a low-pass filter; the magnitude of the tap outputs diminish as the tap number increases. Therefore the convergence speed will strongly depend on each individual tap.

Further, we reset only one of the weight to zero after the system converged. Figure 4.11 shows a comparison of the gamma filter and Laguerre filter in terms of convergence speed. Note, the convergence speed depends on the individual weight. However we can use w_0 to represent the average convergence speed. For the system ID problem, the Laguerre filter converges faster than the gamma filter. This result is

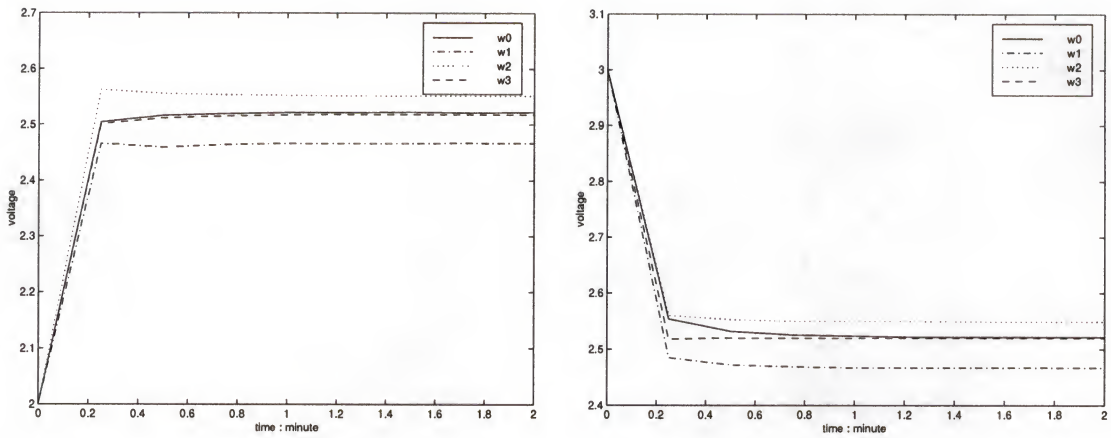


Figure 4.10: Weights track with different initial values. (a) Initial weights equal 2.0 V. (b) Initial weights equal 3.0 V.

consistent with the orthogonality property of the Laguerre filter. In Figure 4.11(a), the weights which were initialized to zero (2.5 V) showed a fast convergence speed. In other words, adapting all weights simultaneously at the beginning can provide a faster convergence than adapting only one weight leaving the others at stable situation.

In summary, the Laguerre filter has the advantage of faster convergence

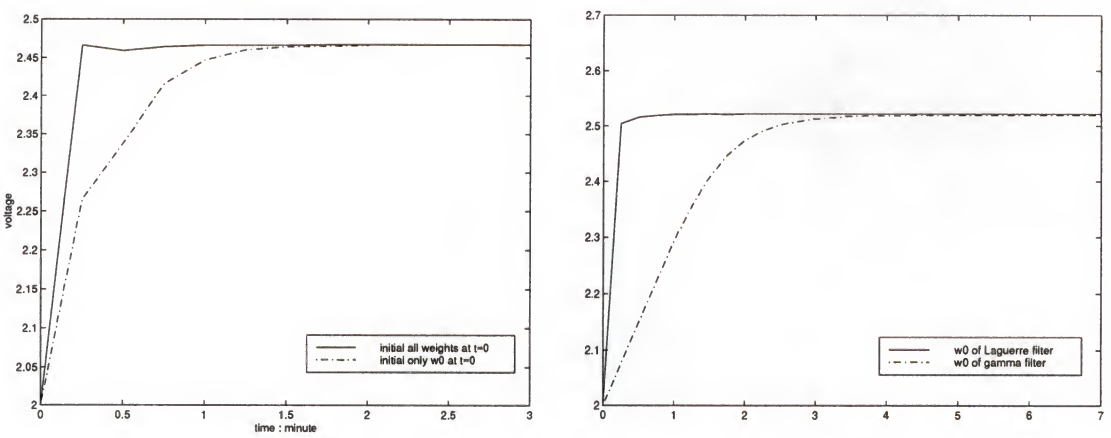


Figure 4.11: w_0 weight track comparison: (a) comparison on Laguerre filter with initialized all weights and initialized one weight after stable. (b) comparison weight tracks between the Laguerre and gamma filters.

speed than the gamma filter but needs more chip space and worse DC offset than the gamma filter.

CHAPTER 5
MULTI-SCALE CONCEPT

As discussed in previous chapters, the gamma and Laguerre memory are superior to the standard tap delay line because of their ability to automatically choose an appropriate time-scale [21, 40, 39]. We have not taken advantage of this capability in the previous chapters since the time constant of each stage of the gamma and Laguerre networks was held constant. An adaptive time constant becomes particularly significant for problems involving extremely long impulse responses for which the standard tap delay line solutions can require thousands of taps. Unfortunately, the gamma and Laguerre structures have a few problems of their own:

1. Choosing the optimal time scale is a nonlinear optimization problem. Gradient descent is not guaranteed to find the optimal time scale. This problem becomes particularly troublesome when we build dedicated hardware (analog or digital) for implementing these filters.

2. The adaptive parameters is not a smooth function of the input signal. That is, small perturbations in the signal characteristic can cause large changes in adaptive parameters. This means that the gamma filter is problematic when used to create representations for feature extraction and recognition problems.

3. Even when a single optimal time-scale can be found, the structure cannot efficiently represent information occurring at other time scales.

4. As with the FIR case, choosing an appropriate number of taps is also a difficult optimization procedure.

To deal with these problems, we introduce the multi-scale time constant concept for the memory structure of the generalized feedforward filter. The multi-scale idea can be used in the gamma and Laguerre filters. We derive an analytic solution for the multi-scale gamma (ms-gamma) kernel. The time of peak value of the ms-gamma is equally spaced on a log time plot. We show that the ms-gamma memory can be easily implemented using an additional resistive line. The simulation and chip measurements of a integrated ms-gamma filter are shown and compared to the standard gamma filter. A multi-scale time constant Laguerre filter is introduced. The ms-Laguerre kernels posses the same orthogonal property as the Laguerre kernels. We prove this orthogonality and show the circuit implementation of the ms-Laguerre filter.

5.1 Continuous-time Multi-scale Gamma Filter

First, we introduce a new generalized feedforward filter which is the multiple time-scale extension of the gamma memory (ms-gamma). Multi-scale extensions for discrete-time structures are also very promising and are discussed in Chapter 6. The generalized feedforward filter is shown once again in Figure 5.1. We first need to determine the transfer functions $H_k(s)$ for the ms-gamma.

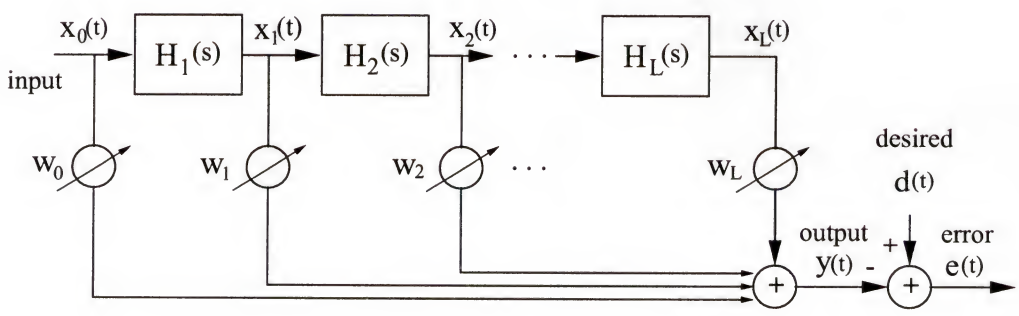


Figure 5.1: The generalized feedforward filter.

Our continuous-time multi-scale gamma structure is shown in Figure 5.2. The ms-gamma is a cascade of first-order low-pass filters with time constants that slow down exponentially as signals propagate down the cascade. If we define the time constant of the last stage to be τ, then the next to the last stage has a time constant of $a\tau$ where $0 < a < 1$. Since the time constant changes by a factor of a for each stage, if we set $a = 1$, the ms-gamma reduces to the usual gamma memory. We can simplify the mathematical analysis by considering an infinite cascade of sections. In general, for $a < 1$

$$H_k(s) = \frac{1}{a^k \tau s + 1} \tag{5.1}$$

The full system response at tap i is given by

$$G_i(s) = \prod_{k=i}^{\infty} \frac{1}{a^k \tau s + 1} \tag{5.2}$$

Because of the infinite cascade, the following scaling laws can be easily derived. In the s-domain:

$$\begin{aligned} G_i(as) &= \prod_{k=i}^{\infty} \frac{1}{a^k \tau (as) + 1} = \prod_{k'=i+1}^{\infty} \frac{1}{a^{k'} \tau s + 1} \\ &= G_{i+1}(s) \end{aligned} \tag{5.3}$$

In the time-domain, the relation between adjacent ms-gamma tap becomes:

$$g_{i+1}(t) = \frac{1}{a} g_i(\frac{t}{a}) \tag{5.4}$$

Therefore the impulse functions $g_i(t)$ are all identical with the exception of a scaling of the amplitude and time axis. We can derive an analytic form of the impulse response at each tap for the ms-gamma for both the finite and infinite cascade versions. Both expressions consist of a weighted sum of exponentials. We can use the partial fraction method for nonrepeated roots to derive the analytic resolution of $g_i(t)$ in the infinite cascade version. The partial fraction equation states that for a transfer function with

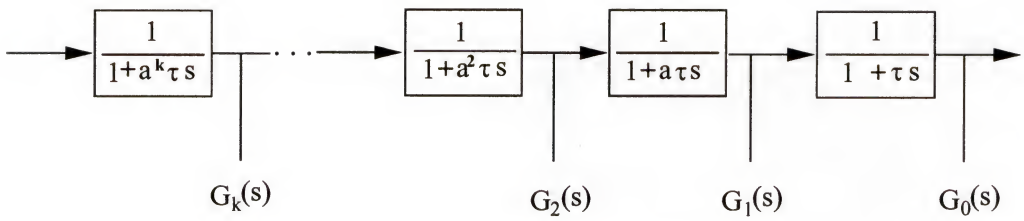


Figure 5.2: Continuous-time multi-scale gamma memory.

N nonrepeated poles, $X(s)$, can be expressed as the summation of N individual pole functions as:

$$X(s) = \frac{1}{Q(s)} = \prod_{k=1}^{N} \frac{1}{s + p_k} = \sum_{k=1}^{N} \frac{A_k}{s + p_k} \tag{5.5}$$

then the inverse transform becomes:

$$x(t) = \sum_{k=1}^{N} A_k e^{-p_k t} \tag{5.6}$$

where

$$A_k = (s + p_k) \frac{1}{Q(s)} |_{s=-p_k} \tag{5.7}$$

for our ms-gamma kernels, from Equation (5.2) we can get

$$g_i(t) = \sum_{k=i}^{\infty} \frac{A_{k-i}}{a^k \tau} e^{\frac{-t}{a^k \tau}} \tag{5.8}$$

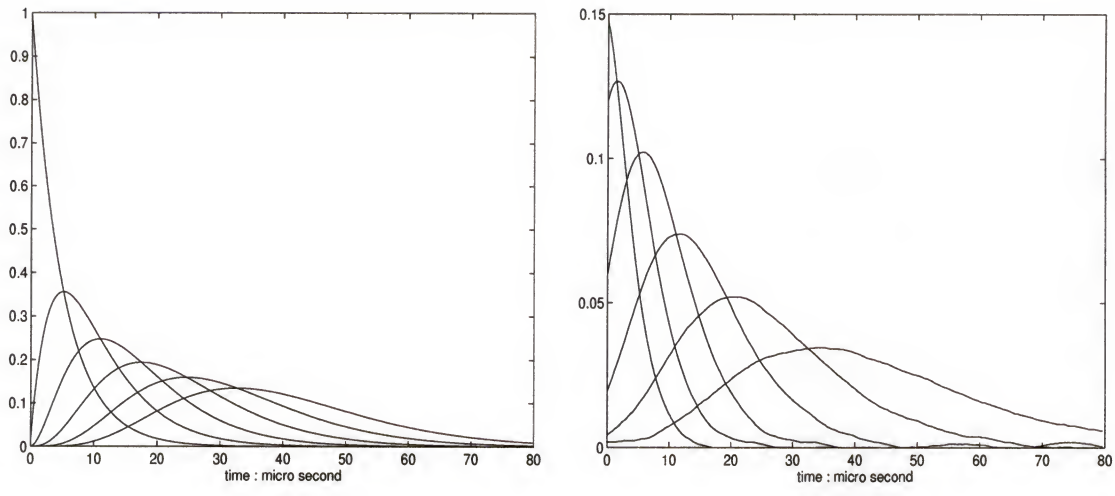
where

$$A_m = \prod_{\substack{j=0 \\ j \neq m}}^{\infty} \frac{1}{1 - a^{j-m}} \tag{5.9}$$

Figure 5.3 shows the simulated and measured impulse response curves of ms-gamma memory.

The peak values of the impulse response are equally spaced for the gamma filter but are not equally spaced for the ms-gamma. For the infinite cascade, define t_i to be the peak value of the impulse response $g_i(t)$, the scaling law in Equation(5.4) results in the following relation:

$$t_{i+1} = a t_i \tag{5.10}$$



(a) (b)

Figure 5.3: (a) Simulated multi-scale gamma kernels. (b) Measured kernels.

which means that the time of the peak at stage $i+1$ is simply the product a and the time of the peak at stage i. This implies that the peak value of the impulse response for consecutive taps of the infinite cascade are equally spaced on a log time plot. Figure 5.4 shows the peak location of 10 normalized tap responses on a log time plot for the finite ms-gamma memory. Notice that after the first few taps, the peak values become equally spaced and the impulse response shapes converge to the same shape. This is exactly what is expected from the infinite cascade analysis.

5.1.1 Circuit Implementation

In order to implement the multi-scale filter, a resistive line is connected along the tap bias controls to achieve a linear voltage drop from one end to the other. Because the CMOS transamps are operated in the sub-threshold region, the output currents are exponentially in the bias voltages, which means their poles also exponentially decreasing as k increases. Figure 5.5 shows the schematic for the L-tap ms-gamma structure. The resistors are chosen with the same value, the factor τ is controlled by the voltage V_{high} and a is set by V_{low}. An eight-tap ms-gamma structure has been fabricated using MOSIS 2μm N-WELL technology. The measured impulse responses from the chip are shown in Figure 5.3(b).

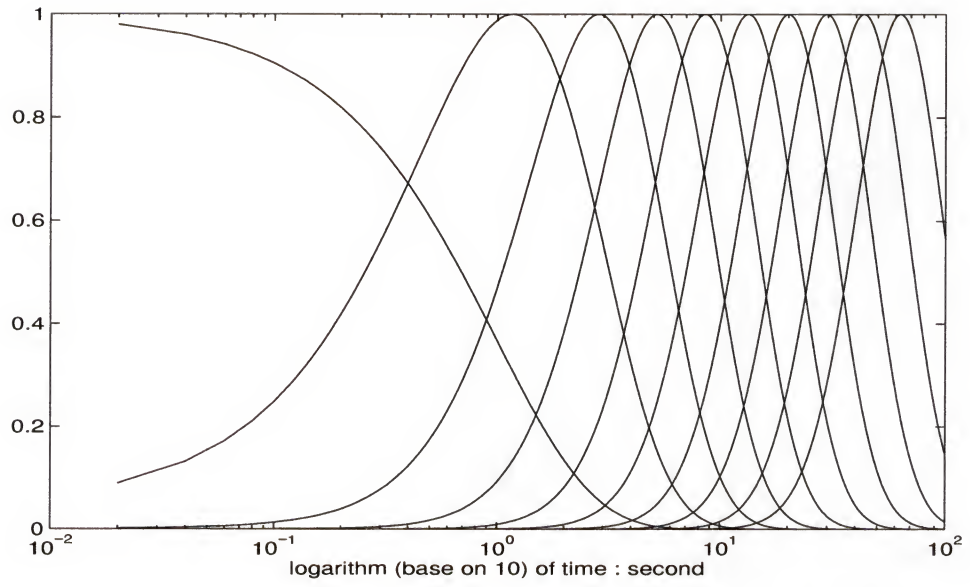


Figure 5.4: Impulse responses of the multi-scale gamma memory on a log-time plot. The peak value of each impulse response has been normalized to unity for display purposes.

5.1.2 Simulations and Measurement Results

We have also performed system identification simulations using the ms-gamma filter in continuous-time. We define a performance index $\xi \equiv E[e^2(t)]$ where e(t) = d(t) - y(t) is an error signal. With a fixed value of parameter a on ms-gamma, we exclusively search the optimal time constant, τ, both for gamma and ms-gamma. Figure 5.6 shows that the ms-gamma performance index is fairly flat when $1/\tau$ is larger than 800. Since the performance surface is fairly constant and close to its global minimum in this region, finding the optimal τ is probably not so important for this structure. Figure 5.7 shows that when the number of taps (L) increases, the performance index becomes even more flat. With our example, the unknown system is a 5^{th} order system and using a 5^{th} order ms-gamma filter is enough to approximate the unknown system. Increasing the number of taps to $L = 6$ does not provide much improvement. These results suggest that there may be no need to find an optimal scale (as is necessary for the gamma) if many time scales are explored simultaneously (as in the ms-gamma).

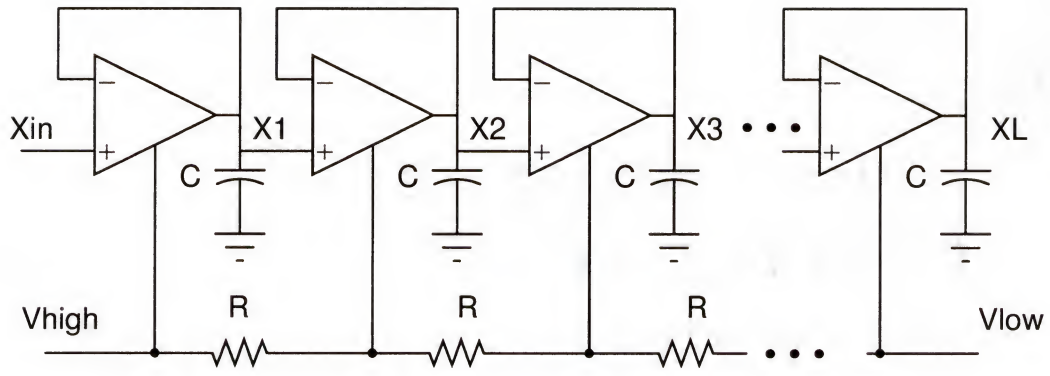


Figure 5.5: Circuit implementation of the multi-scale gamma structure.

Though the best performance can be achieved by adapting the number of

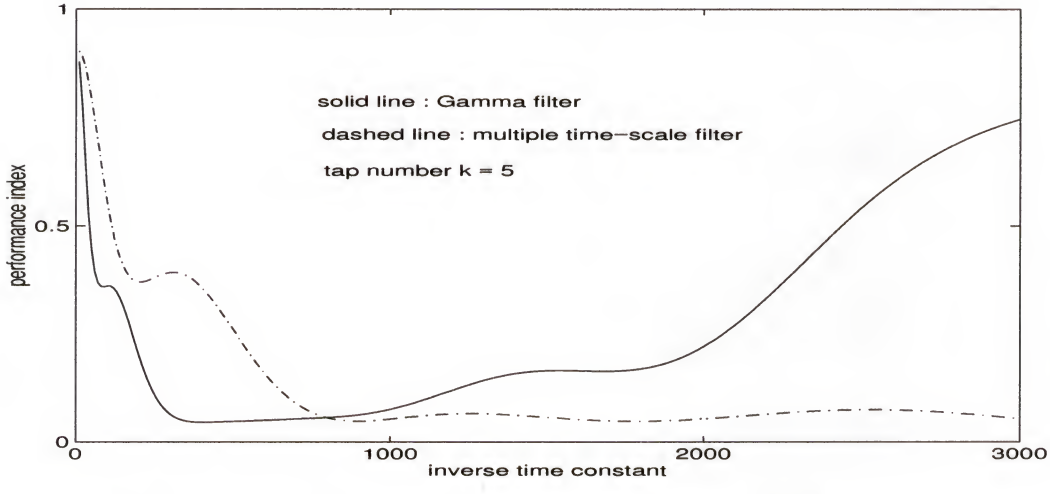


Figure 5.6: Performance index comparison of the gamma and multi-scale gamma filters.

taps, τ, and a, the adaptive procedure would require problematic search procedures and complex hardware. We prefer the following strategy for setting these parameters in real applications:

- The number of taps can be estimated by providing more than enough taps and later pruning those taps that do not contribute significantly to the output. Such a strategy has been previously explored [17].

- The results shown in Figure 5.6 and Figure 5.7 show that the performance of the filter does not depend very strongly on the exact value of $(1/\tau)$ as long as

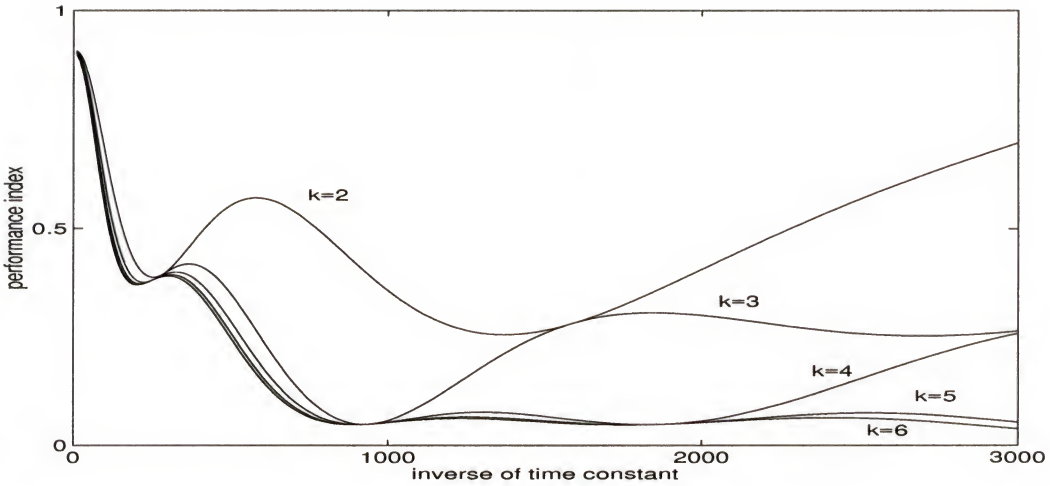


Figure 5.7: Performance index of multi-scale gamma with different numbers of taps (L).

there are enough taps and $1/\tau$ is set high enough. Rather than scanning all possible values of $1/\tau$, we can achieve reasonable performance by setting $1/\tau$ to correspond to an estimate of the highest frequency in the input signal. In the digital version of the ms-gamma discussed in Chapter 6, the fastest frequency was set by the sampling rate.

- The value of a was set to 0.9 in all of our system ID simulations. More research is necessary to discover techniques to set a for different problems. An adaptive strategy is possible but probably not desirable for hardware implementation.

We fabricated an eight-tap ms-gamma filter. When the V_{high} and V_{low} in ms-gamma memory circuit are set equal, then the ms-gamma filter reduces to the standard gamma filter. We test the same system ID problem with ms-gamma and gamma filter. The system ID problem is exactly that discussed in Chapter 3. Figure 5.8 shows the ms-gamma tap outputs while Figure 5.9, and Figure 5.10 show the gamma tap outputs for two different time constants. Comparing the ms-gamma and gamma tap outputs, it is clear to see that the ms-gamma tap outputs have different time constants while the gamma tap outputs have a similar time constant. Figure 5.11

shows the initial output signal and converged output signal. The initial output signal is not zero because of the offset noise. When the unknown system changes in the system ID problem, the gamma filter needs to search for a new optimum time constant. On the other hand, for the ms-gamma filter there is no need to adapt a time constant. Figure 5.12 shows the performance indices of the gamma filter for different unknown systems. The best performance of the standard gamma filter occurs when the bias voltages equal to 0.81 V and 0.83 V, respectively. These best performance index are compared to the ms-gamma filter. Although the ms-gamma performance is slightly worse than the optimal gamma filter, the ms-gamma achieves excellent results without performing a nonlinear search the optimal time constant every time the unknown system changes.

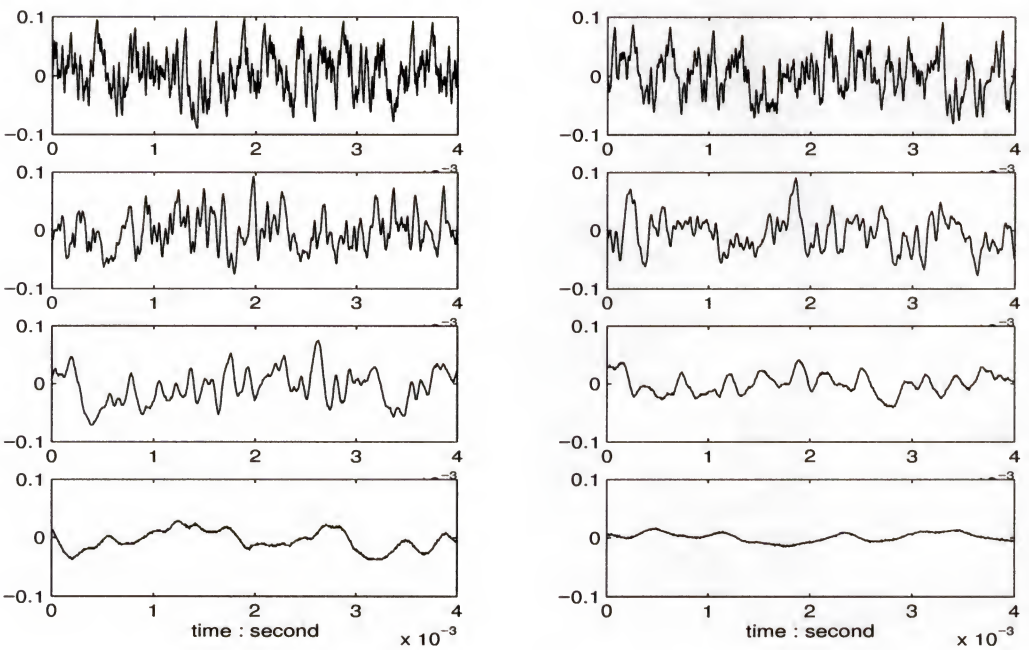


Figure 5.8: The tap outputs of ms-gamma filter. The first tap output is shown in top left, the second tap output is shown in top right. The 8^{th} tap is shown in bottom right.

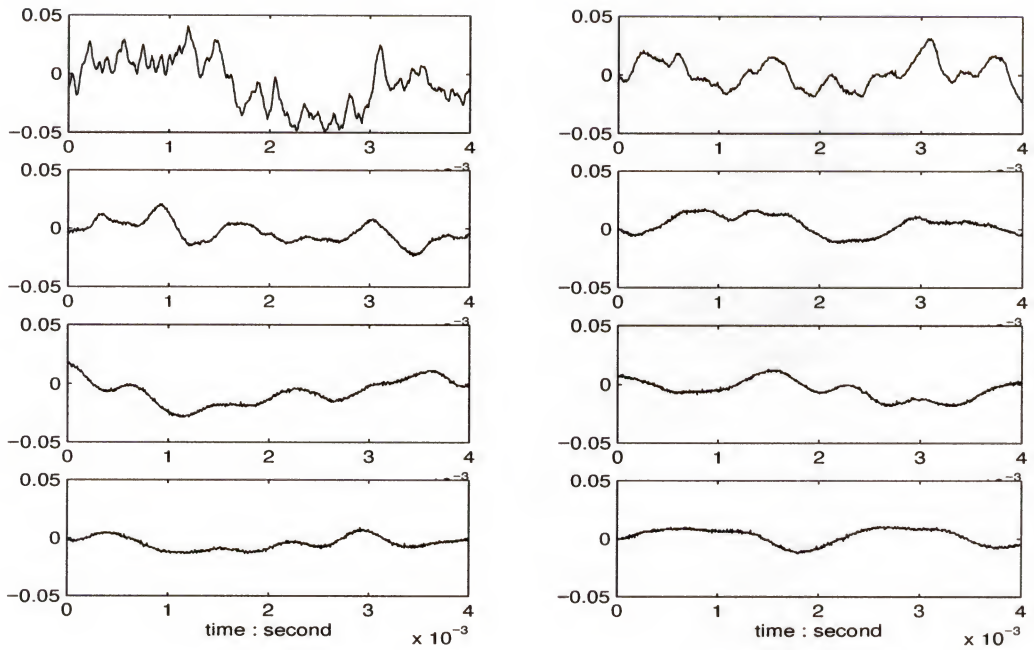


Figure 5.9: The tap outputs of gamma filter with the bias voltage equal 0.65 V. The first tap output is shown in top left, the second tap output is shown in top right. The 8^{th} tap is shown in bottom right.

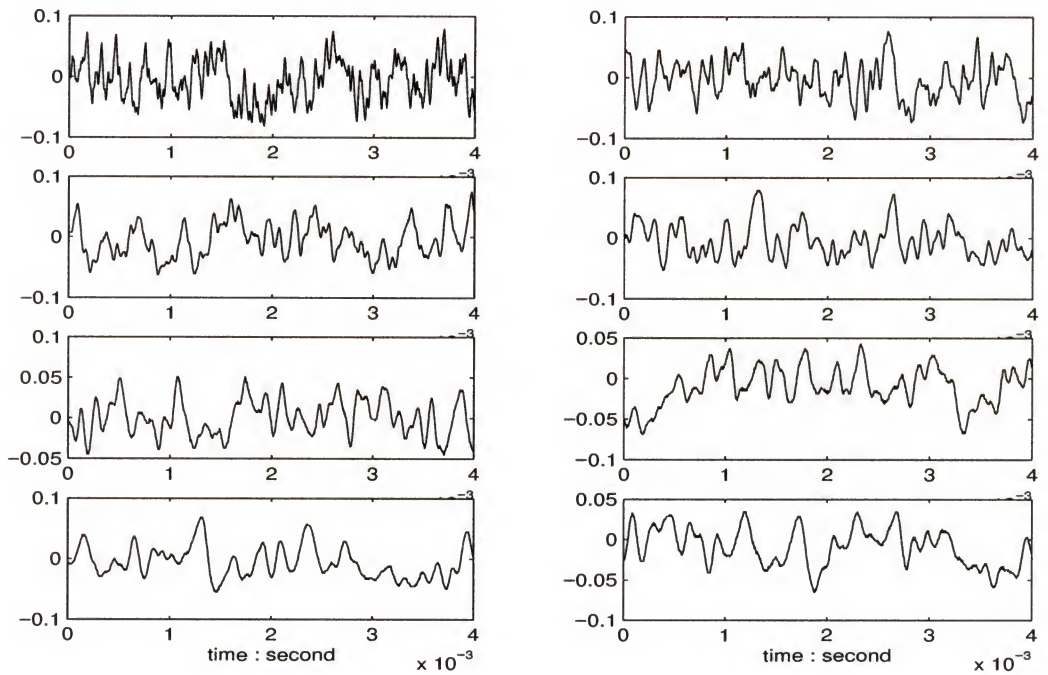


Figure 5.10: The tap outputs of gamma filter with the bias voltage equal 0.75 V. The first tap output is shown in top left, the second tap output is shown in top right. The 8^{th} tap is shown in bottom right.

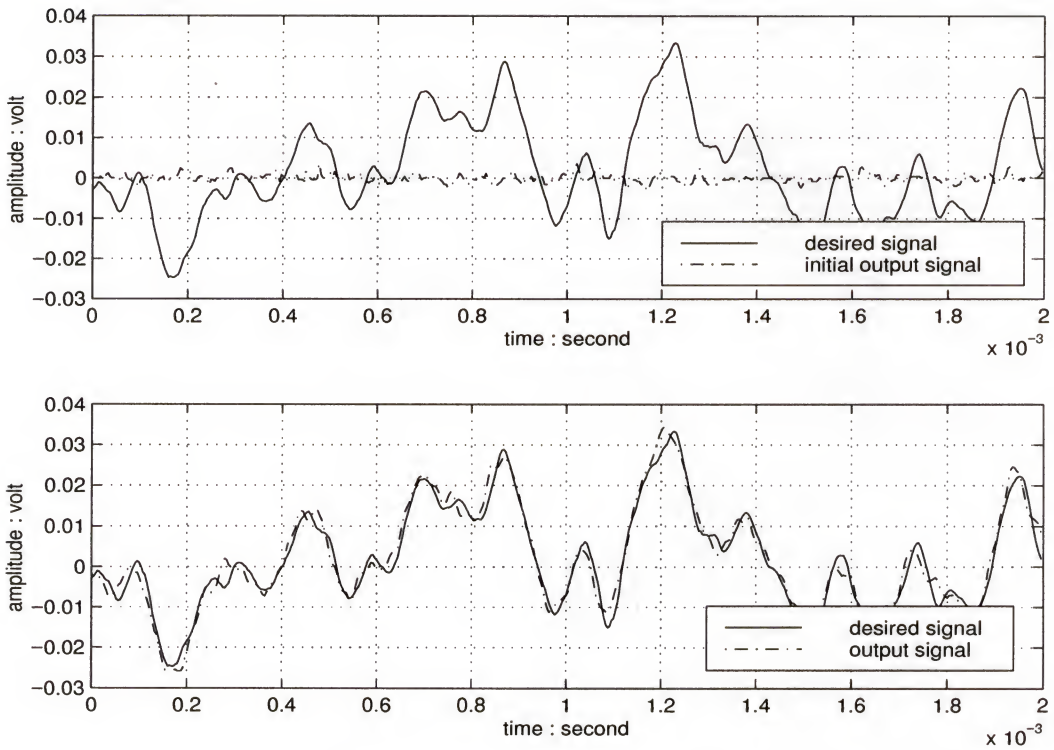


Figure 5.11: The output signal and desired signals. Top, at $t = 0$. Bottom, after convergence.

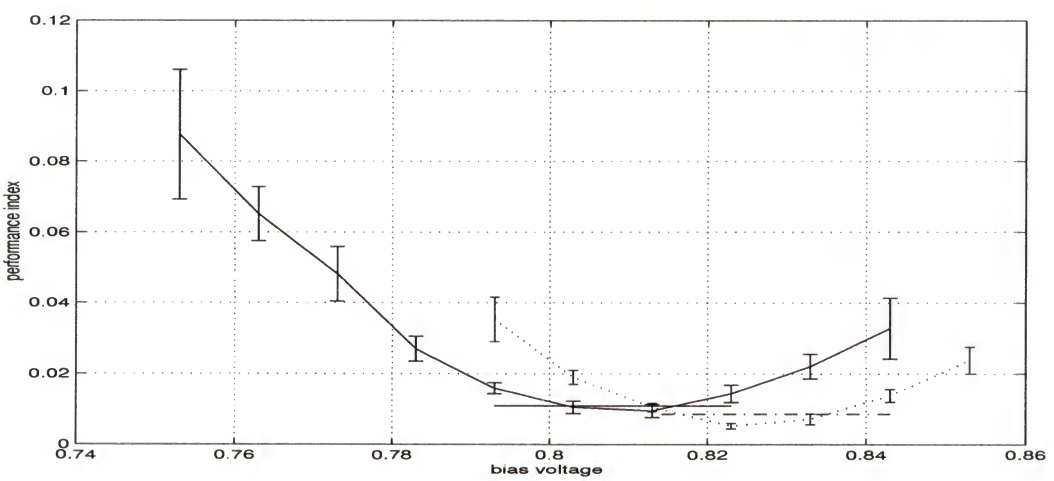


Figure 5.12: Performance index of the ms-gamma and gamma with two different transfer functions. Straight lines show the performance index of ms-gamma filter while error bar curves show the performance of the gamma filter in two different unknown systems.

5.2 Multi-scale Laguerre Kernels

The multi-scale concept can be directly applied to the Laguerre filter also. However, in order to make the ms-Laguerre "compatible" to the Laguerre filter, we require that the ms-Laguerre should also have orthogonal kernels as in the Laguerre. Naively cascading first-order all-pass filter with exponentially varying time constants does not satisfy the orthogonality requirement. We propose an orthogonal ms-Laguerre structure shown in Figure 5.13. The transfer function of the ms-Laguerre therefore can be expressed as:

$$M_k(s) = \frac{1}{1+\tau s}\frac{1-\tau s}{1+a\tau s}\cdots\frac{1-a^{k-1}\tau s}{1+a^k\tau s} \tag{5.11}$$

Because we define the ms-Laguerre structure in the s-domain, it will be simple

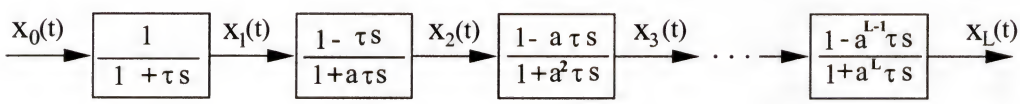


Figure 5.13: Multi-scale Laguerre memory structure.

to prove orthogonality in this domain rather than to transfer to the time domain. However, we start by defining an orthogonal function in the time domain:

$$\int_0^\infty u_m(t)\cdot u_n(t)dt = \begin{cases} 1 & m=n \\ 0 & m\neq n \end{cases} \tag{5.12}$$

We can apply the Parseval's theorem to the functions $u_m(t)$ and obtain their transforms $U_m(\omega)$. According to this theorem

$$\int_0^\infty u_m(t)\cdot u_n(t)dt = 2\pi\int_{-\infty}^\infty U_m^*(\omega)\cdot U_n(\omega)d\omega \tag{5.13}$$

where $U^*(\omega)$ is the complex conjugate of $U(\omega)$. Combining Equation (5.12) and (5.13), if $u_m(t)$ is orthonormal, it turns out in frequency domain it becomes:

$$\int_{-\infty}^\infty U_m^*(\omega)\cdot U_n(\omega)d\omega = \begin{cases} \frac{1}{2\pi} & if\ m=n \\ 0 & if\ m\neq n \end{cases} \tag{5.14}$$

In order to evaluate a real definite integrals over an infinite interval in Equation (5.14), we use a complex variable integral to solve the infinite integral problem. Since ω covers the interval $[-\infty, \infty]$, and ω is real, one part of the contour must involve the real axis. Then the contour can be closed by means of a semicircle in the upper or the lower half-plane. Integrations along these two paths should yield the same values. Since we require that the integral be taken in the positive (counterclockwise) direction about the contour, we should choose a path which is closed in the upper half-plane. The contour is shown in Figure 5.14, where $z = \omega + j\sigma$. For $-R \leq z \leq R$, $z = \omega$, while on the arc over which $0 \leq \theta \leq \pi$, $z = Re^{j\theta}$. We eventually allow R to approach infinity. Therefore it becomes a complex integration shown as:

$$\oint U_m^*(z) \cdot U_n(z) dz \tag{5.15}$$

We perform the complex integration by the method of residues [19, 25]. For

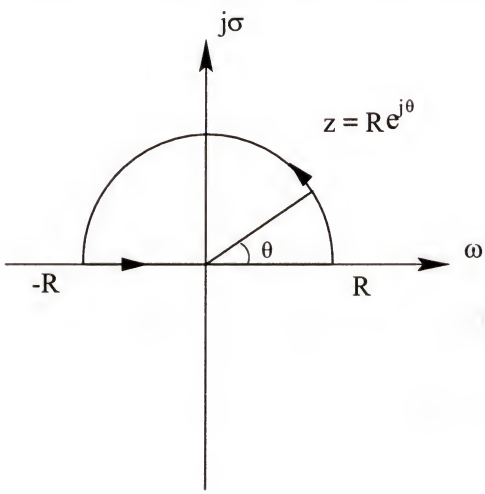


Figure 5.14: A contour for the evaluation of the integral in the upper-half complex plane.

the ms-Laguerre, when $m = n$, equation (5.15) becomes:

$$\oint M_m^*(z) \cdot M_m(z) dz \tag{5.16}$$

$$= \oint \left(\frac{1}{1+\tau jz}\frac{1-\tau jz}{1+a\tau jz} \cdots \frac{1-a^{m-1}\tau jz}{1+a^m\tau jz}\right)^* \cdot \left(\frac{1}{1+\tau jz}\frac{1-\tau jz}{1+a\tau jz} \cdots \frac{1-a^{m-1}\tau jz}{1+a^m\tau jz}\right) dz$$

Because the contour integral is closed in the upper-plane, only the positive poles will contribute to the residue. After arrangement, there is only one pole in the upper half plane. Therefore the residue of the Equation (5.16) becomes:

$$\oint M_m^*(z) \cdot M_m(z)dz = \frac{1}{1 - a^m jz\tau}|_{z=\frac{j}{a^m\tau}} = \frac{1}{2} \tag{5.17}$$

for $m \neq n$, and $m \leq n$

$$\begin{aligned} & \oint M_m^*(z) \cdot M_n(z)dz \\ = & \oint (\frac{1}{1+\tau jz}\frac{1-\tau jz}{1+a\tau jz}\cdots\frac{1-a^{m-1}\tau jz}{1+a^m\tau jz})^* \cdot (\frac{1}{1+\tau jz}\frac{1-\tau jz}{1+a\tau jz}\cdots\frac{1-a^{n-1}\tau jz}{1+a^n\tau jz})dz \\ = & \oint (\frac{1}{1-a^{n-m}\tau jz}\cdot\frac{1}{1-a^{n-m+1}\tau jz}\cdots\frac{1}{1-a^n\tau jz})dz \\ = & \ 0 \end{aligned} \tag{5.18}$$

Equation (5.18) shows there are (n-m+1) poles located in the lower-plane. Since there are no poles in the upper-plane, the residue and the contour integral are both zero. Combining the results of Equation (5.17) and (5.18) we prove the ms-Laguerre produces orthogonal kernels.

5.2.1 Multi-scale Laguerre Circuit Implementation

In order to implement the ms-Laguerre filter, we analyze the circuit shown in Figure 5.15. The transfer function of the circuit can be derived as:

$$\frac{V_{out}}{V_{in}} = \frac{1-\tau s}{1+a\tau s} \tag{5.19}$$

where $a = \frac{R1}{R2}$ is the time constant and $\tau = aRC$ is the geometric ratio.

For each cascaded block of the ms-Laguerre, the ratio of the zero location to the pole location is equal to a. We can implement the ms-Laguerre by cascading the circuit shown in Figure 5.15. By proper connecting the bias voltages, we can obtain the ms-Laguerre circuit shown in Figure 5.16. Here each resistor is replaced by a transamp whose transconductance is controlled by a voltage provided by a linear resistive line.

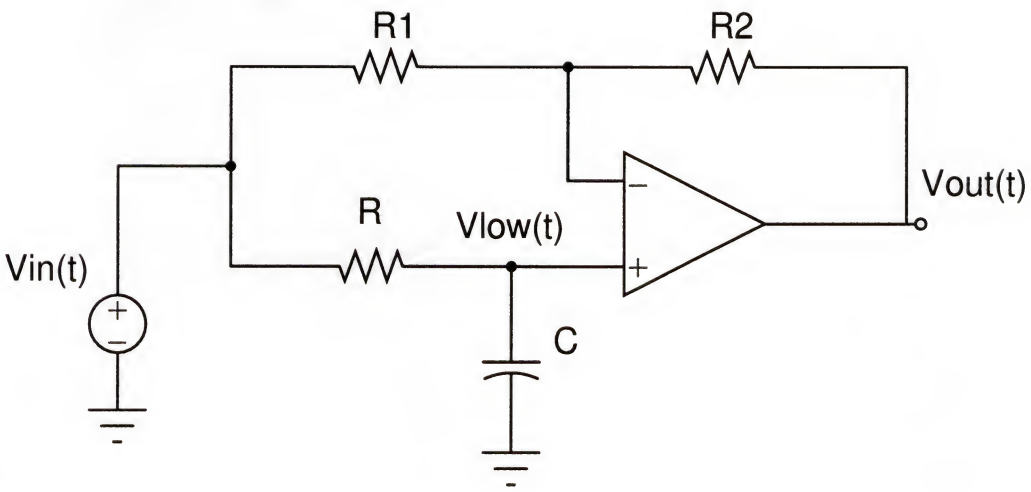


Figure 5.15: Multi-scale Laguerre single tap circuit.

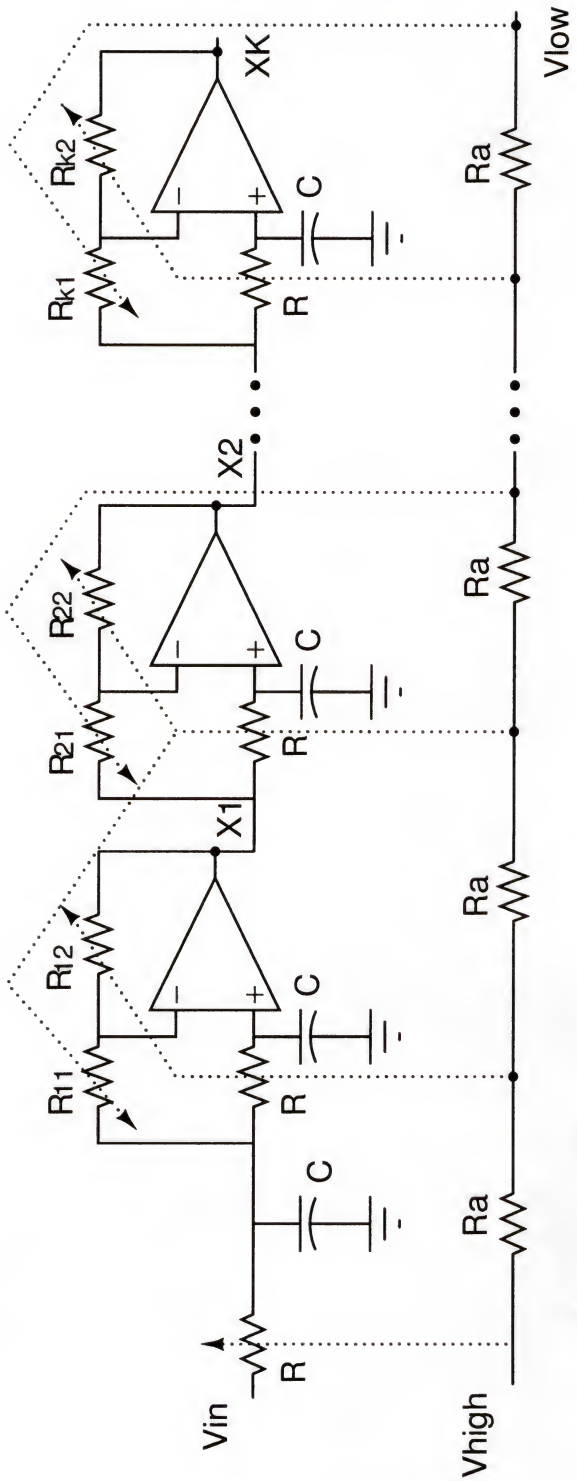


Figure 5.16: Multi-scale Laguerre circuit. The resistors with the same dashed line is biased at the same voltage.

CHAPTER 6
MULTI-SCALE DISCRETE-TIME ADAPTIVE FILTER

In Chapter 2, we derived the properties of continuous-time transversal filters and compared them to their discrete-time counterparts. These results apply to the continuous-time gamma, Laguerre, and multi-scale gamma/Laguerre filters discussed in Chapters 3, 4, and 5 respectively. In each chapter, we not only showed the characteristics of each filter but also demonstrated their performance with different real and simulated examples. Although these ideas are based in the analog domain, they can be directly applied to the digital domain. This chapter will explore the multi-scale concept in the discrete-time domain. First we will review the characteristics of the discrete-time gamma and Laguerre filters. The discrete-time gamma and Laguerre are well studied in [41, 42, 35]. The discrete-time multi-scale gamma (ms-gamma) will be introduced and applied to different problems. These results are compared with the standard single-time gamma and Laguerre.

6.1 Single Time Scale Gamma/Laguerre Filters

The discrete-time systems of interest are typically sampled-data system where data is collected from sampling the input/output signals of a continuous-time system. Therefore the discrete-time gamma can be easily derived from its continuous-time counterpart. First we rewrite the continuous-time gamma first-order differential equation in (3.10)

$$\frac{dx_k(t)}{dt} = -\mu x_k(t) + \mu x_{k-1}(t) \tag{6.1}$$

The derivative in Equation (6.1) can be approximated using a first-order forward difference, that is

$$\frac{dx_k(t)}{dt} \approx x_k(n) - x_k(n-1) \tag{6.2}$$

where we assume a unit time step. Substituting Equation (6.2) into (6.1), the discrete-time difference equation is given as

$$x_k(n) = (1-\mu)x_k(n-1) + \mu x_{k-1}(n-1) \qquad k = 1, ..., L \tag{6.3}$$

The tap outputs of an L-tap gamma filter can be calculated recurrently from the previous taps based on Equation (6.3). Also from Equation (6.3) the gamma unit, $G(z)$, which represents the transfer function between adjacent taps, can be derived as:

$$G(z) = \frac{X_k(z)}{X_{k-1}(z)} = \frac{\mu}{z-(1-\mu)} \tag{6.4}$$

Now, the output signal of the gamma filter can be written as:

$$\begin{aligned} Y(z) &= \sum_{k=0}^{k=L} w_k[G(z)]^k X(z) \\ y(n) &= \sum_{k=0}^{k=L} w_k x_k(n) \end{aligned} \tag{6.5}$$

The discrete-time gamma filter can be realized as shown in Figure 6.1, where $x_0(n) = x(n)$ is the input signal.

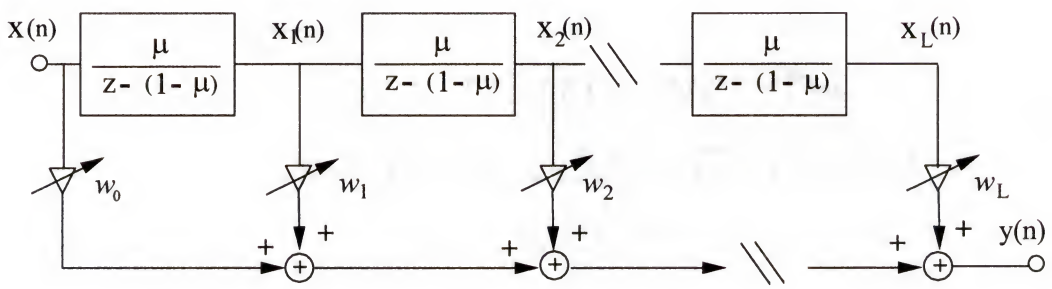


Figure 6.1: Discrete-time gamma filter.

As in the continuous-time domain, the power of the gamma filter is that its unique time constant provides a tradeoff between memory depth and resolution. The

unique repeated pole of the gamma occurs at:

$$z_p = (1 - \mu) \tag{6.6}$$

As a result, the μ value must satisfy the condition

$$0 < \mu < 2 \tag{6.7}$$

to guarantee stability. The time constant of the discrete-time gamma is quite different from its continuous-time counterpart. In discrete-time, the μ is confined to the region $0 \rightarrow 2$. For the special case of $\mu = 1$, the gamma becomes an FIR transversal filter. On the contrary, in the continuous-time domain, the time constant is related to its 3 dB cutoff frequency. Theoretically, there is no limit of the time constant in the continuous-time. Also there is no value in continuous-time that will transform the system to an FIR filter.

Again the LMS algorithm is chosen to update the weights of the gamma filter. The adaptation rule is based on the gradient signal

$$w_k(n+1) = w_k(n) + \eta_w e(n) x_k(n) \tag{6.8}$$

and the discrete-time update rule for μ can be written as:

$$\mu(n+1) = \mu(n) + \eta_\mu \sum_{k=0}^{L} e(n) w_k(n) \alpha_k(n) \tag{6.9}$$

where $\alpha_k(n)$ is the gradient signal defined by

$$\alpha_k \equiv \frac{\partial x_k(n)}{\partial \mu} \tag{6.10}$$

leading to the following iterative procedure:

$$\begin{aligned} \alpha_0(n) &= 0 \\ \alpha_k(n) &= (1-\mu)\alpha_k(n-1) + \mu\alpha_{k-1}(n-1) + x_{k-1}(n-1) - x_k(n-1) \\ k &= 1, ..., L \end{aligned} \tag{6.11}$$

The form of the Wiener-Hopf solution for the gamma filter is identical to that of an FIR transversal filter except now the correlation matrix, $\mathbf{R}_\mu$ and the cross-correlation vector, $\mathbf{P}_\mu$ are computed by the tap outputs, $x_k(n)$, instead of the delay of the input signal $x(n-k)$ as in the FIR filter case.

The L-tap Laguerre filter can be realized as in Figure 6.2.

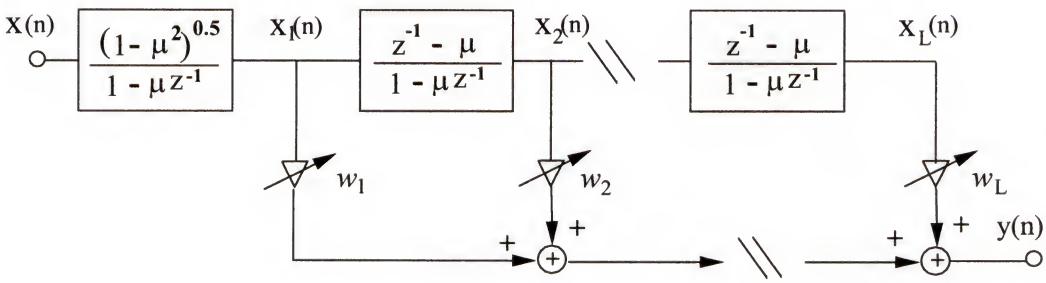


Figure 6.2: Discrete-time Laguerre filter.

The z-transforms of these sequences are given by

$$L_k(z) = \sqrt{1-\mu^2}\frac{(z^{-1}-\mu)^k}{(1-\mu z^{-1})^{k+1}}, \quad ,k \geq 0 \tag{6.12}$$

Further, for $k \geq 0$ these cascaded first-order all-pass filter can be expressed as

$$L_{k+1}(z) = A(z)L_k(z) \tag{6.13}$$

where $A(z)$ represents the first tap all-pass filter given by

$$A(z) = \frac{z^{-1}-\mu}{1-\mu z^{-1}} \tag{6.14}$$

Like the gamma filter, the Laguerre filter has a unique pole which is equal to μ [1]. In order to guarantee stability, the pole, μ, must satisfy

$$-1 < \mu < 1 \tag{6.15}$$

[1]There are different expressions for the μ between the gamma and the Laguerre. For the gamma, the pole is equal to $1-\mu$ while the pole of the Laguerre is equal to μ.

For an L-tap Laguerre filter, the output signal is:

$$y_k(n) = \sum_{k=0}^{L} w_k x_k(n) \tag{6.16}$$

Where the k^{th} tap output is the convolution of the input signal and $L_k(\mu)$.

$$x_k(n) = L_k(\mu) * x_0(n) \tag{6.17}$$

The first tap output of the Laguerre filter is identical to that in the gamma unit except for a scaling

$$\begin{aligned} X_1(z) &= \frac{\sqrt{1-\mu^2}}{1-\mu z^{-1}} X_0(z) \\ x_1(n) &= \mu x_1(n-1) + \sqrt{1-\mu^2}\, x_0(n) \end{aligned} \tag{6.18}$$

For the k^{th} tap output, $k > 1$, this becomes

$$\begin{aligned} X_k(z) &= \frac{z^{-1}-\mu}{1-\mu z^{-1}} X_{k-1}(z) \\ x_k(n) &= \mu x_k(n-1) + x_{k-1}(n-1) - \mu x_{k-1}(n) \end{aligned} \tag{6.19}$$

For the LMS algorithm, the update rules of weights and μ are identical to the gamma filter except that the gradient signal for Laguerre is different, expressed as:

$$\begin{aligned} \alpha_0(n) &= 0 \\ \alpha_1(n) &= x_1(n-1) + \alpha_1(n-1) - \frac{\mu}{\sqrt{1-\mu^2}} x_0(n) \\ \alpha_k(n) &= \mu\alpha_k(n-1) + \alpha_{k-1}(n-1) - \mu\alpha_{k-1}(n) + x_k(n) - x_{k-1}(n) \\ k &= 1, ..., L \end{aligned} \tag{6.20}$$

For the Laguerre filter, there exists an important connection between the correlation matrix, $\mathbf{R}_\mu$, and the correlation matrix of a FIR filter [35]. Starting with the Wiener solution of an FIR filter

$$\mathbf{RW} = \mathbf{P} \tag{6.21}$$

where the (m,n) elements of $\mathbf{R}$ and m^{th} element of $\mathbf{P}$ are given

$$\begin{aligned} r_{m,n} &= E[x(k-m)x(k-n)] \\ &= \frac{1}{2\pi}\int_{-\pi}^{+\pi} e^{j\omega(n-m)}\Phi_{xx}(e^{j\omega})d\omega \end{aligned} \tag{6.22}$$

$$\begin{aligned} p_m &= E[x(k)x(k-m)] \\ &= \frac{1}{2\pi}\int_{-\pi}^{+\pi} e^{j\omega m}\Phi_{yx}(e^{j\omega})d\omega \end{aligned} \tag{6.23}$$

$\Phi_{xx}(e^{j\omega})$ and $\Phi_{yx}(e^{j\omega})$ are the input power spectral density and cross power spectral density respectively.

Using the same procedures, the (m,n) element of the correlation matrix of the Laguerre filter $\mathbf{R}_\mu$ can be written as:

$$\begin{aligned} \tilde{r}_{m,n} &= E[x_m x_n] \\ &= E[x(k) * l_m(k) \cdot x(k) * l_n(k)] \\ &= \frac{1}{2\pi}\int_{-\pi}^{+\pi} e^{j\omega(n-m)}\Phi_{x_m x_n}(e^{j\omega})d\omega \end{aligned} \tag{6.24}$$

where '*' is convolution. Replacing the power spectral density, $\Phi_{x_m x_n}(e^{j\omega})$ between x_m and x_n with the power spectral density, $\Phi_{xx}(e^{j\omega})$, between $x(k-m)$ and $x(k-n)$, Equation (6.24) becomes

$$\begin{aligned} \tilde{r}_{m,n} &= \frac{1}{2\pi}\int_{-\pi}^{+\pi} L_m(e^{j\omega})L_n^*(e^{j\omega})\Phi_{xx}(e^{j\omega})d\omega \\ &= \int_{-\pi}^{+\pi} \left(\frac{e^{j\omega}-\mu}{1-\mu e^{j\omega}}\right)^{n-m} \frac{\Phi_{xx}(e^{j\omega})(1-\mu^2)d\omega}{2\pi|1-\mu e^{j\omega}|^2} \end{aligned} \tag{6.25}$$

Using the same method, the m^{th} element of $\mathbf{P}_\mu$ is

$$\begin{aligned} \tilde{p}_m &= E[x(k) \cdot x(k) * *L_m(k)] \\ &= \int_{-\pi}^{+\pi} \left(\frac{e^{i\omega}-\mu}{1-\mu e^{j\omega}}\right)^{m} \frac{\Phi_{yx}(e^{j\omega})\sqrt{1-\mu^2}d\omega}{2\pi(1-\mu e^{j\omega})} \end{aligned} \tag{6.26}$$

It is possible to simplify the expression for $\tilde{r}_{m,n}$ by introducing a new variable, θ. Let

$$e^{j\theta} = \frac{e^{j\omega} - \mu}{1 - \mu e^{j\omega}} \tag{6.27}$$

Actually, if we vary the ω from 0 to 2π, then the change of θ is also in the range 0 to 2π. Next we can invert the relation, expressing ω as a function of θ.

$$e^{j\omega} = \frac{e^{j\theta} + \mu}{1 + \mu e^{j\theta}} \tag{6.28}$$

Differentiating both sides, we get the new integral variable $d\theta$ as

$$d\theta = \frac{1 - \mu^2}{|1 - \mu e^{j\omega}|^2} d\omega \tag{6.29}$$

Inserting (6.28)(6.29) into (6.25), the $\tilde{r}_{m,n}$ can be expressed as

$$r_{m,n} = \frac{1}{2\pi} \int_{-\pi}^{+\pi} e^{j\theta(n-m)} \Phi_{xx}(\frac{e^{j\theta+\mu}}{1 + \mu e^{j\theta}}) d\theta \tag{6.30}$$

A comparison of (6.30) with the FIR correlation matrix in (6.22) shows both $r_{m,n}$ and $\tilde{r}_{m,n}$ have the same format; they only differ in the augmentation of the power spectral density. This result can be used to determine immediate bounds for the eigenvalues of $\mathbf{R}_\mu$ based on the matrix $\mathbf{R}$ in the FIR transversal filter.

6.2 Multi-scale Gamma Filter

The multiple time constant concept is proposed in order to avoid the adaptation of the single time scale for the gamma/Laguerre filters. The reasons are similar to the continuous-time case.

1. Choosing the optimal time scale is a nonlinear optimization problem. Gradient descent is not guaranteed to find the optimal time scale.

2. The adaptive parameter are not a smooth function of the input signal. That is, small perturbations in the signal characteristic can cause large changes in adaptive parameters. This means that the gamma filter is problematic when used to create representations for feature extraction and recognition problems.

3. Even when a single optimal time-scale can be found, the structure cannot efficiently represent information occurring at other time scales.

4. Choosing an appropriate number of taps is also a difficult optimization procedure.

Our proposed discrete-time multi-scale gamma filter is shown in Figure 6.3. Unlike the gamma filter, the location of the pole at each stage depends on the tap k. The transfer function between taps can be written as:

$$\frac{x_k(z)}{x_{k-1}(z)} = \frac{a^{k-1} \cdot \mu}{z - (1 - a^{k-1} \cdot \mu)} \tag{6.31}$$

where a is the geometric ratio and we assume that $a < 1$. If we perform a two-dimensional search for the optimal a and μ, this structure cannot perform worse than the original gamma, since the gamma filter is included as a special case when $a = 1$. We now compare the performance of the gamma, the Laguerre, and the ms-gamma filter for different problems.

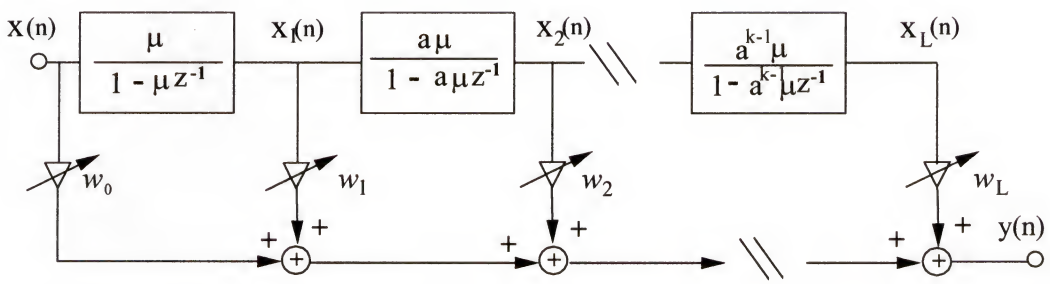


Figure 6.3: Discrete-time multi-scale gamma filter.

6.3 Simulations

6.3.1 System Identification

We demonstrate a problem for which the multi-scale filter easily outperforms the gamma filter by posing a system ID problem that includes two widely separated poles. We use the transfer function H(z):

$$H(z) = \frac{0.3}{(z - 0.05)(z - 0.95)} \tag{6.32}$$

For the gamma structure, we scan all possible values of μ between 0 and 1. For the multi-scale gamma structure we set $\mu = 1$ so that the first stage is exactly an ideal delay. We then scan all values of of a between 0 and 1. The Wiener-Hopf equations were used to solve for the optimal weight values in order to obtain the MSE [41]. The results for various numbers of taps (from 2 to 5) are shown in Figure 6.4. In all cases, the gamma filter has more difficulty in approximating the system with a small number of taps. Since we are scanning all values of the free parameter in both cases, a practical optimization procedure is still an open problem. The solution for the multi-scale gamma can be further improved if we also optimize for μ instead of setting $\mu = 1$. We did not explore this direction because we seek simple search methods that can be readily implemented in dedicated hardware.

The multi-scale gamma structure is now better able to represent signals with widely varying time constants but we still require a difficult optimization procedure to find the optimal a (assuming $\mu = 1$). Rather than performing this difficult search procedure, we borrow a standard weight pruning technique from neural network theory [8]. We purposely include more taps than we need in order to cover a large range of time scales and selectively deactivate any weight values that are small in magnitude.

We have simulated a system ID problem using the multi-scale gamma filter in which the transfer function of the unknown system is given by:

$$H(z) = \frac{0.4(z - 1)}{z(z - 0.1)(z - 0.7)(z - 0.9)} \tag{6.33}$$

The Mean Square Error vs. number of taps is shown in Figure 6.5. For an L-tap filter, the smallest $10 - L$ weight magnitudes are set to zero. We again assume that $\mu = 1$. The are a few things to note in Figure 6.5. First, the $a = 0.9$ solution is

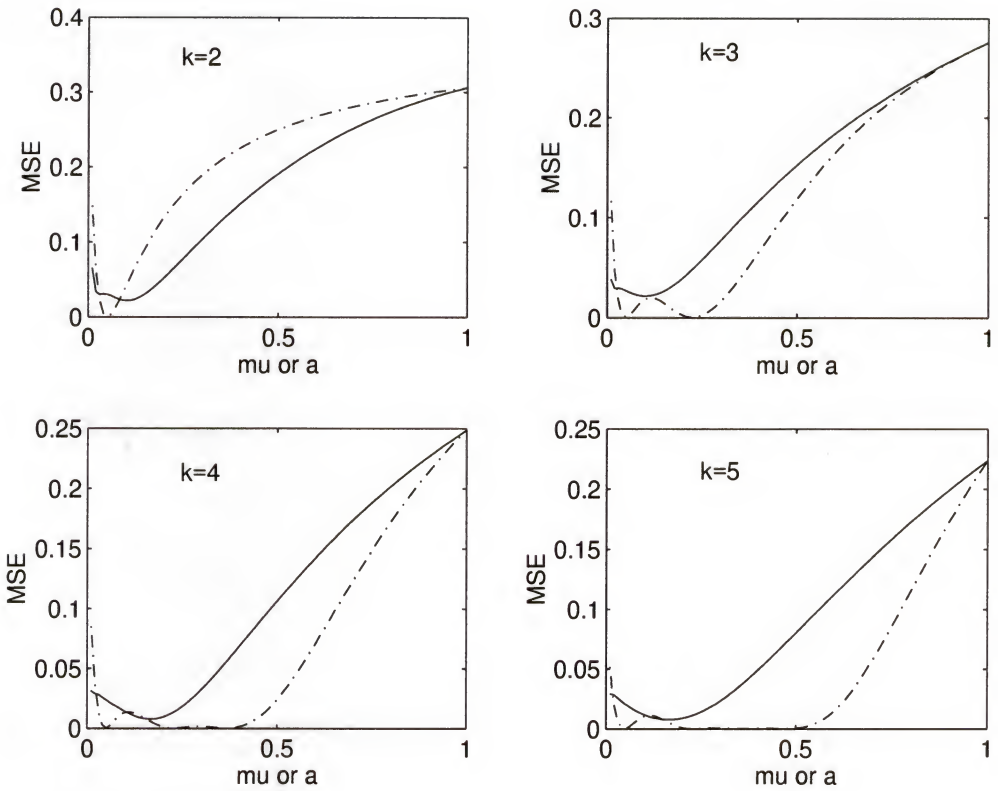


Figure 6.4: Mean square error comparisons between the gamma filter (solid line) and the multi-scale gamma filter (dotted line). We plot MSE vs μ (for gamma) or a for (multi-scale gamma).

better than $a = 0.5$ for an equal number of taps. For $a = 0.5$, the poles are spaced too far apart. Second, for both values of a there is a sharp transition beyond which adding more taps does not decrease the error. This sharp transition can be used to choose a reasonable number of taps for each problem. Minimizing the number of taps reduces the overall amount of computation and also lowers the misadjustment of the system. We plan to use this weight pruning method to avoid implementing complex, non-convex optimization procedures. The weight pruning method also provides a mechanism for choosing an appropriate number of taps [27].

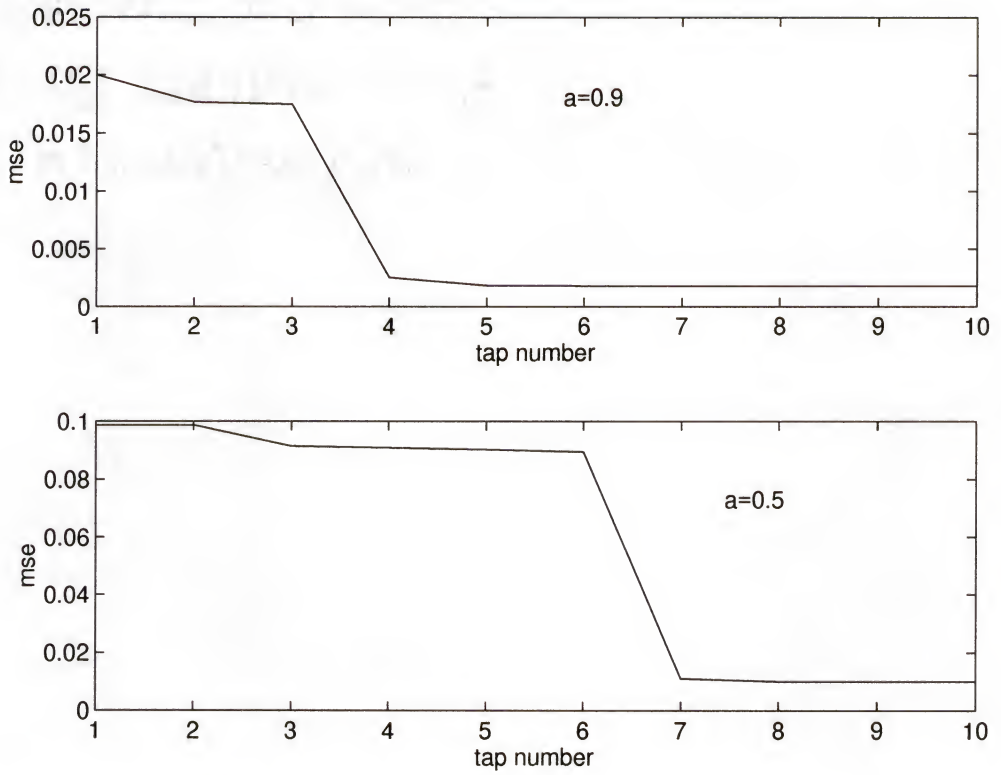


Figure 6.5: Plot of MSE vs. number of taps using the weight pruning method. For each number of taps, the smallest magnitude weights are set to zero.

6.3.2 Echo Cancellation

Both voice and data transmission over telephone channels are impaired by the echo generated in the hybrid circuits that perform the conversion adaptation between two and four lines. This type of echo problem can be effectively suppressed with adaptive filters. In order to obtain different sequences of binary data for the transmitter, we input Gaussian random noise to the hard limiter shown in Figure 6.6. The output of the hard limiter is binary data, ± 1.

We choose two different echo transfer functions to compare the performances among the gamma, the Laguerre and the ms-gamma. First we choose a third-order repeated pole transfer function, $H(z)$, as:

$$H(z) = \frac{0.1(1 + z^{-1})(-1 + z^{-1})^2}{(1 + 0.3z^{-1})^3} \tag{6.34}$$

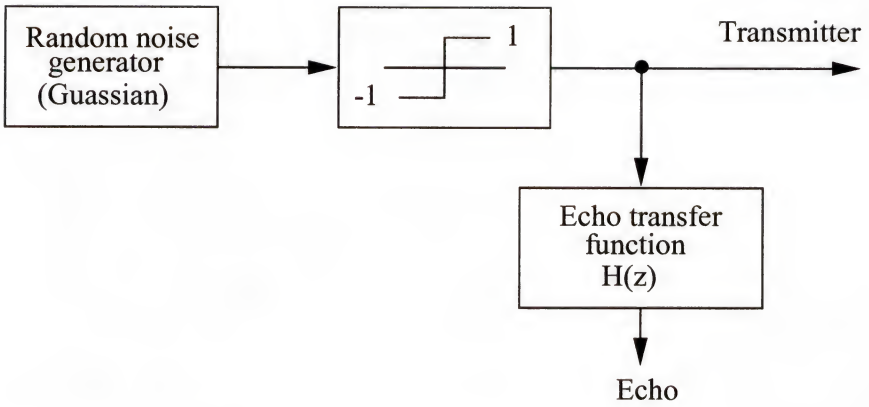


Figure 6.6: Block diagram for generating the echo.

Figure 6.7 shows the MSE of third-order ms-gamma filter as a function of μ and a. Not surprisingly the best performance of ms-gamma filter is when a equals 1 since the unknown system has a single repeated pole like the gamma. Also Figure 6.8(a) shows the gamma filter outperforms the Laguerre filter in this case.

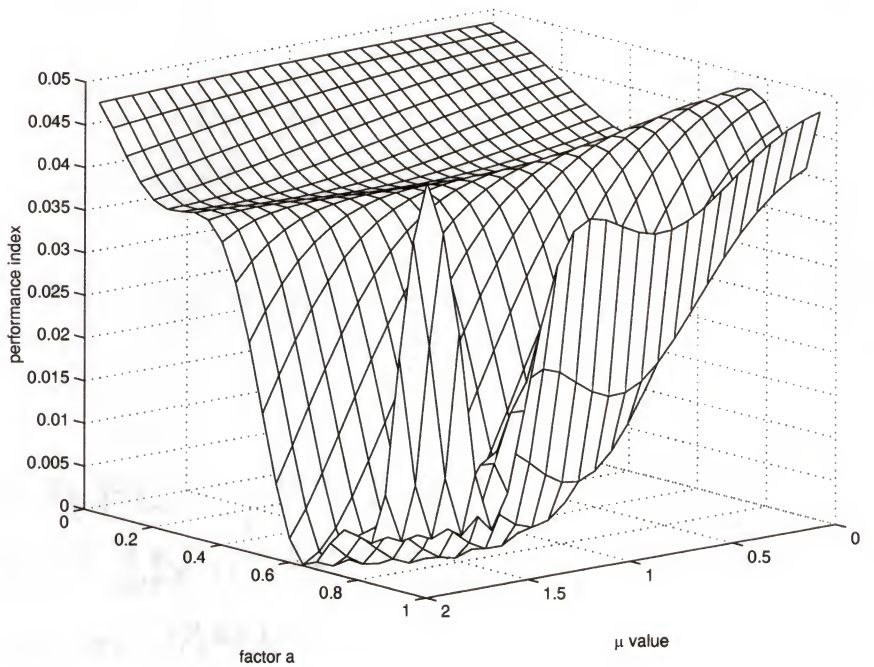
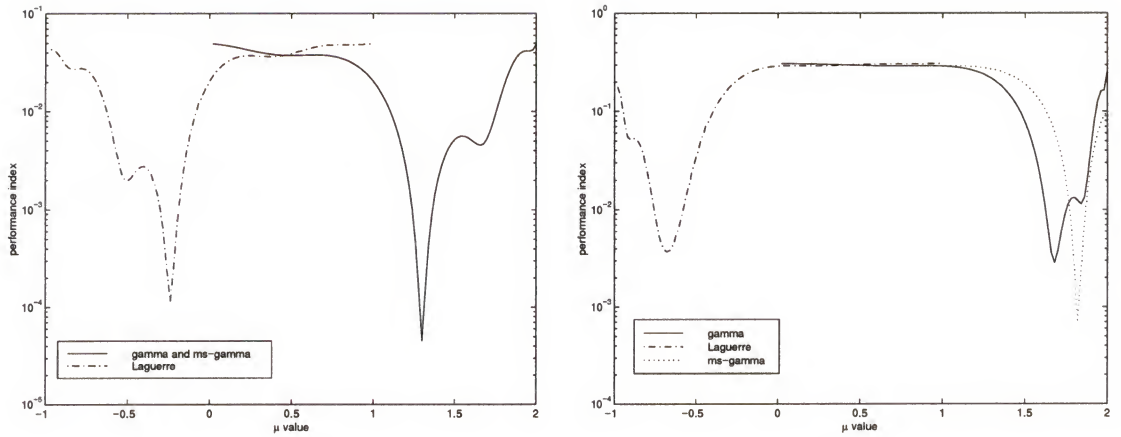


Figure 6.7: The ms-gamma performance index respective to μ and a.

If we consider a third-order non-repeated pole transfer function as:

$$H(z) = \frac{0.1(1+z^{-1})(-1+z^{-1})^2}{(1+0.2z^{-1})(1+0.7z^{-1})(1+0.8z^{-1})} \tag{6.35}$$



(a) (b)

Figure 6.8: (a) MSE comparison for repeated pole echo transfer function, (b) MSE comparison for nonrepeated pole echo transfer function.

Table 6.1: MSE comparison for the gamma and ms-gamma

order	min MSE(gamma)	min MSE(ms-gamma)	optimal a	MSE(a=0.8)
3	0.0142	0.0104	0.44	0.1265
4	0.0124	0.0094	0.57	0.0378
5	0.0083	0.0074	0.67	0.0143
6	0.0069	0.0068	0.72	0.0073
7	0.0068	0.0067	0.77	0.0067
8	0.0007	0.0003	0.83	0.0003

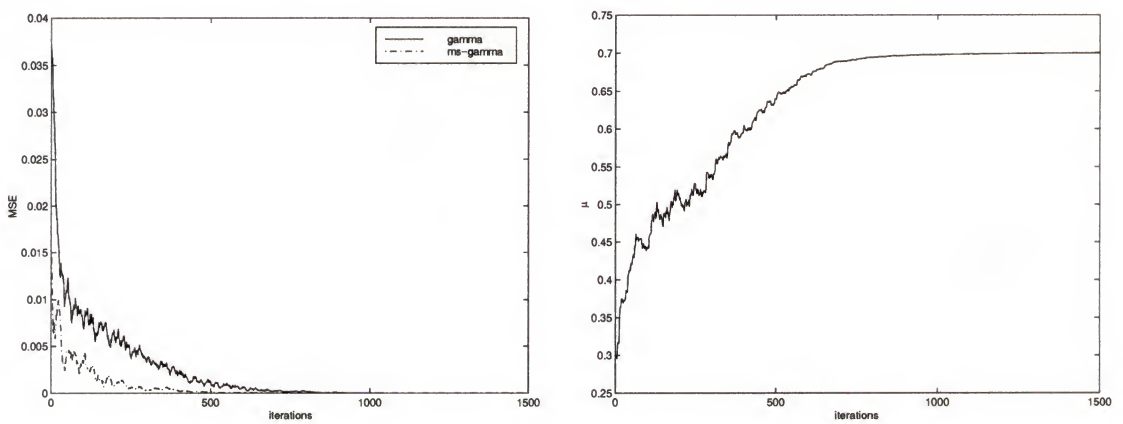
Figure 6.8(b) shows the performances of the gamma, Laguerre, and the ms-gamma. Note, the ms-gamma performance is setting to its optimal a value. In here, the ms-gamma performs better then the gamma for the non-repeated pole case. However, the search for the optimal μ and a in the ms-gamma filter is time consuming. In order to make the ms-gamma useful, we set the μ always equal to 1 and only search the optimal a for different tap orders. Table 6.1 summarizes the optimal MSE for different orders of the gamma and the ms-gamma as well as the optimal a and the MSE for $a = 0.8$ in the ms-gamma.

From Table 6.1, the optimal value of a in the ms-gamma increases as the order increases, therefore if the ms-gamma filter order is large enough (i.e., order = 7), then the performance of $a = 0.8$ in the ms-gamma is compatible to a 4^{th}-order gamma

filter.

Now we use the LMS method to solve the echo cancellation problem by setting the μ and a in the ms-gamma equal to 1 and 0.8 respectively. We compare the learning curves of the 3^{rd}-order gamma filter to the 7^{th}-order ms-gamma for the repeated pole echo transfer function. Figure 6.9(a) shows the learning curves of the gamma and the ms-gamma averaged from 30 ensembles, the μ convergence curve of the gamma is shown in Figure 6.9(b). These results suggest that fixing μ and a to proper values, the high-order ms-gamma filter can achieve the same performance as a lower-order gamma filter.

We have shown that the multi-scale extension of the gamma filter is interesting in the discrete-time domain even though the ideas originated in the continuous-time domain. The discrete-time multi-scale Laguerre can also be derived in a straight forward fashion.



(a) (b)

Figure 6.9: (a) The learning curves comparison of 3^{th}-order gamma and 7^{th}-order ms-gamma. (b) The convergence curve of μ for the gamma filter.

CHAPTER 7
CONCLUSIONS AND FUTURE WORK

The projects described in this dissertation demonstrate the utility of analog VLSI technology for implementing continuous-time transversal adaptive filters. We start with the developed a rigorous analysis of transversal type continuous-time adaptive filters including the issues of proper time constants, misadjustment, and convergence rate. We derived the Wiener equation of the continuous-time transversal filter. We showed that there is no upper bound for the learning rate of continuous-time adaptive filters using gradient descent method. In discrete-time an upper bound is set by the maximum eigenvalue of $\mathbf{R}_d$. We also showed in both discrete and continuous-time cases, the convergence time is set by the reciprocal of the product of the learning rate and the minimum eigenvalue. The LMS algorithm is quite useful for practical problems, we showed the adaptation rule for the weights and give a example to show the tradeoff between the learning rate and misadjustment.

Based on Chapter 2, we implement three continuous-time adaptive filters:

The Gamma filter is a cascaded first-order low-pass filter. The unique time constant provides the freedom of choosing proper tap resolution and memory depth for different problems. The gamma filter has been proven to outperform the FIR transversal filter in many applications, especially in case that requires long impulse responses. We can choose a large memory depth by decreasing the tap resolution without increasing the number of tap in the gamma filer while the FIR transversal filter may need thousands of taps to achieve the same performance. The gamma filter is a special type of IIR filter with only one repeated pole. This constraint allows the stability of the gamma to be easily controlled. We implemented the continuous-time

gamma filter with 2μ CMOS technology, all the circuits are operated in sub-threshold region in order to take the advantage of wide input range and low power consumption. We used the leaky LMS algorithm instead of the LMS algorithm in order to keep the circuits in their linear range of operation. We tested the gamma filter on system ID and noise canceling problems. The results showed the gamma filter can effectively solve many problems.

The Laguerre filter is promising because it guarantees that the impulse response of the taps are orthogonal to one another. We also implemented the continuous-time Laguerre filter in analog VLSI. Simulation and measured results demonstrated an improved performance over the gamma filter with faster convergence rate and higher S/N ratio. The disadvantage of the Laguerre filter is that it needs about four times fabrication area than the gamma filter.

The multi – Scale gamma/Laguerre filters have been introduced for the first time. These multi-scale extensions get rid of the complex nonconvex searches for an optimal time constant. By varying the time constant exponentially down the cascade, many time constants are included simultaneously. We implemented both the ms-gamma and the ms-Laguerre. Measured results for the ms-gamma chip were shown for a system ID problem.

Chapter 6 explored the multi-scale concept in the discrete-time domain. The discrete-time ms-gamma can be directly realized from its continuous-time counterpart. We showed a example to demonstrate the ms-gamma filter can easily outperform the gamma filter by posing a system ID problem that includes two widely separated poles. We also compared the results among the gamma, Laguerre, and ms-gamma for the echo cancellation problem. We conclude that if the order of the ms-gamma filter is large enough, then we can fix $\mu = 1$ and choose a fixed proper geometric ratio, a, which can achieve the same performance as a lower order standard gamma filter in this case.

There are still some unanswered questions. Throughout this thesis we have used off-chip capacitors to store the values of the weights. This led to full control of the measurement and convergence rate of the weights by changing the size of the capacitors. At this point, the capacitors need to be integrated onto the chip. We also must consider the use of floating-gate storage mechanisms so that the values of the weights can be stored for longer periods of time. We have only just begun to explore our multi-scale concepts for adaptive filters. We now need to fully test our ideas on more large-scale applications for both the continuous- and discrete-time implementations.

APPENDIX A
DC Offsets in LMS Algorithm

In order to explore the excess MSE caused by DC offsets, first both the input and error signal are assumed to be generated from zero-mean Gaussian distributions. Even though this assumption is not generally true, it may reasonably be inferred that similar results could be shown for other distributions. Second, a new notation will be introduced for convenience. For a specific time, i, the time index, t, can be dropped for these continuous-time equations and be replaced by the symbol i. Starting with the output signal, the equation (3.19), becomes

$$y(i) = \sum_{k=0}^{L} w_k(i) \cdot x_k(i) \tag{A.1}$$

where L is the order of filter. The two column vectors $\mathbf{W}_i = [w_0(i), w_1(i), ..., w_L(i)]^T$ and $\mathbf{X}_i = [x(i), x_1(i), ..., x_L(i)]^T$ are used for shorthand. Now the output signal can be rewritten as the product of two vectors

$$y(i) = \mathbf{X}_i^T \cdot \mathbf{W}_i \tag{A.2}$$

Given the desired signal, $d(i)$, at time i, the error signal $e(i)$ becomes

$$\begin{aligned} e(i) &= d(i) - y(i) \\ &= \mathbf{X}_i[\mathbf{W}^{*T} - \mathbf{W}_i^T] \end{aligned} \tag{A.3}$$

Here, $\mathbf{W}^*$ represents the optimal weight values. We define a weight deviation vector, $\mathbf{V}_i$, which represents the difference between the optimal weights and current weights

$$\mathbf{V}_i = \mathbf{W}^* - \mathbf{W}_i \tag{A.4}$$

where $\mathbf{V}_i$ also is a column vector, substituting $\mathbf{V}_i$ into Equation (A.3), the error signal becomes

$$e(i) = \mathbf{X}_i \mathbf{V}_i^T \tag{A.5}$$

According to our previous assumption, the input signal and error signal are zero mean. So

$$E[\mathbf{X}_i] = 0 \qquad \text{and} \qquad \mathrm{E}[e(i)] = 0 \tag{A.6}$$

In order to discuss the effect of offsets, there are two types of DC offsets that must be added into the model. The first DC offset occurs in each tap output signals, $[x_0, x_1, ..., x_L]$, and is represented by a vector $\mathbf{m}_x = [m_0, m_1, ..., m_L]^T$. The $\mathbf{m}_x$ is assumed to be constant and independent of time. The second DC offset to be modeled is the unwanted DC offset in the error signal, m_e, which is a scalar value. Now the update equation of the weight vector using the LMS algorithm, shown in equation (3.23), can be remodeled with the following discrete update:

$$\mathbf{W}_{i+1} = \mathbf{W}_i + 2\eta_w[(\mathbf{X}_i + \mathbf{m}_x)(e(i) + m_e)] \tag{A.7}$$

Accordingly, Equation (A.4) the update rule becomes

$$\mathbf{V}_{i+1} = \mathbf{V}_i - 2\eta_w[(\mathbf{X}_i + \mathbf{m}_x)(e(i) + m_e)] \tag{A.8}$$

Taking the expectation of each side gives:

$$E[\mathbf{V}_{i+1}] = E[\mathbf{V}_i] - 2\eta_w E[(\mathbf{X}_i + \mathbf{m}_x)(e(i) + m_e)] \tag{A.9}$$

As $i \to \infty$, the system converges, which means $E[\mathbf{V}_{i+1}] = E[\mathbf{V}_i]$. After arrangement the equation can be simplified to

$$E[\mathbf{X}_i\mathbf{X}_i^T]E[\mathbf{V}_i] = -m_e\mathbf{m}_x \tag{A.10}$$

Now the expected value of $\mathbf{V}_i$ can be expressed as

$$E[\mathbf{V}_i] = -\mathbf{R}^{-1}(m_e\mathbf{m}_x) \tag{A.11}$$

where $\mathbf{R} \equiv E[\mathbf{X}_i\mathbf{X}_i^T]$ is the auto-correction of $\mathbf{X}_i$.

In order to measure error variance, σ_e^2, we need to go back to Equation (A.9)

and take the mean square value of both sides:

$$\begin{aligned} E[\mathbf{V}_{i+1}^T\mathbf{V}_{i+1}] &= E[\mathbf{V}_i^T\mathbf{V}_i] - 4\eta_w E[\mathbf{V}_i^T(\mathbf{X}_i+\mathbf{m}_x)(e(i)+m_e)] \\ &+ 4{\eta_w}^2 E[(\mathbf{X}_i+\mathbf{m}_x)(e(i)+m_e)^T(\mathbf{X}_i+\mathbf{m}_x)(e(i)+m_e)] \end{aligned} \tag{A.12}$$

When the system reaches steady-state, the variance of $\mathbf{V}_{i+1}$ and $\mathbf{V}_i$ is the same, $E[\mathbf{V}_{i+1}^T\mathbf{V}_{i+1}] = E[\mathbf{V}_i^T\mathbf{V}_i]$. This equation can be simplified as:

$$E[\mathbf{V}_i^T(\mathbf{X}_i+\mathbf{m}_x)(e(i)+m_e)] = \eta_w E[(\mathbf{X}_i+\mathbf{m}_x)(e(i)+m_e)^T(\mathbf{X}_i+\mathbf{m}_x)(e(i)+m_e)] \tag{A.13}$$

The solution of Equation (A.13) with arbitrary η_w is tedious and results in a value for the MSE that has a weak dependence on η_w. The right-hand side can be dropped showing:

$$E[\mathbf{V}_i^T(\mathbf{X}_i+\mathbf{m}_x)(e(i)+m_e)] = 0 \tag{A.14}$$

By expanding the multiplication, we get the summation of four terms equaling 0.

$$E[\mathbf{V}_i^T\mathbf{X}_i e(i)] + E[\mathbf{V}_i^T\mathbf{X}_i m_e] + E[\mathbf{V}_i^T\mathbf{m}_x e(i)] + E[\mathbf{V}_i^T\mathbf{m}_x m_e] = 0 \tag{A.15}$$

Because $e(i) = \mathbf{X}_i^T\mathbf{V}_i$, the first term is exactly the excess MSE due to the DC offsets

$$E[\mathbf{V}_i^T\mathbf{X}_i e(i)] = E[e(i)^T e(i)] = \sigma_e^2 \tag{A.16}$$

According to the assumptions we made in the beginning, the expectation of $\mathbf{X}_i$ is zero, $E[\mathbf{X}_i] = 0$, and using the property of the independence between $\mathbf{V}_i$ and $\mathbf{X}_i$, the second term becomes

$$E[\mathbf{V}_i^T\mathbf{X}_i m_e] = E[\mathbf{V}_i^T\mathbf{X}_i]m_e = E[\mathbf{V}_i^T]E[\mathbf{X}_i]m_e = 0 \tag{A.17}$$

The third term also equals zero based on the assumption of $E[\mathbf{X}_i] = 0$

$$\begin{aligned} E[\mathbf{V}_i^T\mathbf{m}_x e(i)] &= E[\mathbf{V}_i^T\mathbf{m}_x\mathbf{X}_i^T\mathbf{V}_i] \\ &= E[\mathbf{V}_i^T\mathbf{V}_i\mathbf{m}_x\mathbf{X}_i] = 0 \end{aligned} \tag{A.18}$$

To solve the final term, we insert the result derived in Equation (A.11)

$$\begin{aligned} E[\mathbf{V}_i^T\mathbf{m}_x m_e] &= E[\mathbf{V}_i^T]\mathbf{m}_x m_e \\ &= -(\mathbf{m}_x m_e)^T\mathbf{R}^{-T}(\mathbf{m}_x m_e) \end{aligned} \tag{A.19}$$

substituting these terms into Equation (A.15), the variance of error caused be the DC offsets can be represented as

$$\sigma_e^2 \simeq (\mathbf{m}_x \mathrm{m_e})^{\mathrm{T}} \mathbf{R}^{-\mathrm{T}} (\mathbf{m}_x \mathrm{m_e}) \tag{A.20}$$

REFERENCES

[1] Aggarwa J.K., Nonlinear System Stability Analysis, Dowden, Huntchingon and Ross, Inc., Stroudsburg, PA, 1977.

[2] Bult K., Wallinga H., A class of analog CMOS circuits based on the square-law characteristic of an MOS transistor in saturation, IEEE Journal of Solid-State Circuits, 22(3):357-361, June 1987.

[3] Celebi S., Representation of locally stationary signals using lowpass moments, Ph.D. dissertation, University of Florida, 1995.

[4] deVries B. Temporal processing with neural networks: the development of the gamma model, Ph.D. dissertation, University of Florida, 1991.

[5] deVries B., Principe J.C., The gamma neural network-a new model for temporal processing, Neural Networks, 5:565–576, 1992.

[6] Dolenko B., Card H., Tolerance to analog hardware of on-chip learning in back-propagation networks, IEEE Transactions on Neural Network, 6(5):1045–1052, September 1995.

[7] Franco S., Design with Operational Amplifiers and Analog Integrated Circuits, McGraw-Hill, New York, 1988.

[8] Haykin S., Neural networks: A comprehensive foundation, IEEE Computer Society Press, New York, 1994.

[9] Haykin S., Adaptive Filter Theory, Prentice Hall, Englewood Cliffs, NJ, 1996.

[10] Johns D.A., Snelgrove W.M., Sedra A.S., State-space adaptive recursive filters, Proceedings of IEEE Int. Symp. Circuits and System, pages 2153–2156, 1988.

[11] Johns D.A., Snelgrove W.M., Sedra A.S., Continuous-time analog adaptive recursive filters, Proceedings of IEEE Int. Symp. Circuits and System, pages 667–670, 1989.

[12] Johns D.A., Snelgrove W.M., Sedra A.S., Adaptive recursive state-space filters using a gradient-based algorithm, IEEE Transactions on Circuits and Systems, 37(6):673–683, June 1990.

[13] Johns D.A., Snelgrove W.M., Sedra A.S., Continuous-time LMS adaptive recursive filters, IEEE Transactions on Circuits and Systems, 38(7):769–777, July 1991.

[14] Jones D.G., System modeling using generalized linear feedforward networks, Ph.D. research proposal of University of Florida, 1996

[15] Juan J., Harris J.G., and Principe J.C., Analog VLSI implementations of continuous-time memory structures, Proceedings of IEEE Int. Symp. on Circuits and Systems, 3:338–340, 1996.

[16] Juan J., Harris J.G., and Principe J.C., Analog hardware implementations of adaptive filter structures, Proceedings of IEEE International Conference on Neural Networks, 2:916–921, 1997.

[17] Juan J., Harris J.G., and Principe J.C., Locally-recurrent network with multiple time-scales, IEEE Workshop on Neural Networks for Signal Processing, p645-653, 1997.

[18] Juan J., Harris J.G., Analog hardware inplementation of continuous-time adaptive filter structures, Journal of Analog Integrated Circuits and Signal Processing, to appear August 1997.

[19] Karni S., Byatt W.J., Mathematical Methods in Continuous and Discrete Systems, Holt, Rinehart and Winston, New York, 1982.

[20] Kuo J., Celebi S., and Principe J.C., Adaptation of memory depth in the gamma filter, Processing ICASSP94, 5:373–376, 1994.

[21] Kuo J. and Principe J.C., Noise reduction in state space using the focused gamma model, Proc. ICASSP94, 2:533–536, 1994.

[22] Kwan T. and Martin K., An adaptive analog continuous-time CMOS biquadratic filter, IEEE Journal of Solid-State Circuits, 26(6):859–867, June 1991.

[23] Linares-Barranco B., Sanchez-Sinencio E., and Rodriguez-Vazquez A., A CMOS analog adaptive BAM with on-chip learning and weight refreshing, IEEE Transactions on Neural Network, 4(3):445–455,May 1993.

[24] Lebedev V.L., Random processes in electrical and mechanical system, NSF NASA Tech. Translation, office of Technical Services, Washington, DC, 1958.

[25] Lee Y.W., Statistical Theory of Communication, John Wiley and Sons Inc., New York, 1960.

[26] Lont J. and Guggenbuhl W., Analog CMOS implementation of a multi-layer perception with nonlinear synapses, IEEE Transactions on Neural Network, 3(3):457–465, May 1992.

[27] Lozowski A., Miller D., and Zurada J., Dynamics of error back-propagation learning with pruning in the weight space, Proceedings of IEEE Int. Symp. Circuits and Systems, 3:449–452, 1996.

[28] Macchi O., Adaptive Processing: The Least Mean Squares Approach with Applications in Transmission, John Wiley and Sons Ltd, New York, 1995.

[29] Martin K. and Sun M.T., Adaptive filters suitable for real-time spectral analysis, IEEE Transactions on Circuits and Systems, 33(2):218–230, February 1986.

[30] Mayyas K.A., Aboulnasr T., Leaky LMS: A detailed analysis, Proceedings of IEEE Int. Symp. Circuits and Systems, 2:1255-1258, 1995.

[31] Mead C., Analog VLSI and Neural Systems, Addison-Wesley, Reading, MA, 1989.

[32] Moon G., Zaghloul M.E., and Newcomb R.W., VLSI implementation of synaptic and summing in pulse coded neural-type cells, IEEE Transactions on Neural Networks, 3(3):394–403, May 1992.

[33] Oppenheim A.V. and Schafer R.W., Discrete-Time Signal Processing, Prentice Hall, Englewood Cliffs, NJ, 1989

[34] Ovlveira T.S., ON the equivalance between gamma and Laguerre filters, ICASSP p19-22, 1994.

[35] Ovlveira T.S., Laguerre filters - an introduction, Revista do DETUA 1(3):237-248 1995.

[36] Palkar M., Adaptive of the gamma filter for echo cancellation, Master Thesis, University of Florida, 1993.

[37] Palkar M. and Principe J.C., Echo cancellation with the gamma filter, Proc. ICASSP, 3:369–372, 1994.

[38] Principe J.C., Kuo J., and Celebi S., An analysis of short term memory structures in dynamic neural networks, IEEE Transations on Neural Networks, 5(2):331–337, 1994.

[39] Principe J.C., Motter M., System identification with dynamic neural networks, Proc. W. Cong. Neural Networks, 2:284–289, 1994.

[40] Principe J.C., Rathie A., Kuo J., Prediction of chaotic time series with neural networks, In Nonlinear Dynamic of the Brain, p250-258, 1993.

[41] Principe J.C., deVries B., The gamma filter-a new class of adaptive IIR filters with restricted feedback, IEEE Transactions on Signal Processing, 41(2):649–656, February 1993.

[42] Principe J.C., Zahalka A., Transient detection using neural network: the search for the desired signal, Neural Information Processing System, 5:688–695, 1993.

[43] Salam F.M., Oh H.J., Real-time control using nodular neural chip with on-chip learning, In Proceedings of International Conference on Neural Networks, 1996.

[44] Shoval A., Johns D.A., Snelgrove W.M., Comparison of DC offsets in four LMS adaptive algorithm, IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, 42(3):176–185, March 1995.

[45] Snelgrove W.M., Sedra A., Synthesis and analysis of state-space active filters using intermediate transfer functions, IEEE Transactions on Circuits and Systems, 33(3):287–301, March 1986.

[46] Sondhi M., An adaptive echo canceler, Bell Syst. Tech. vol. 50, pp785-813, March 1971.

[47] Strum R.D., Kirk D.E., First Principles of Discrete Systems and Digital Signal Processing, Addison-Wesley, Reading, MA, 1989.

[48] Vittoz E.A., Analog VLSI signal processing: Why, where and how, Journal of VLSI Signal Processing, 8:27–44, July 1994.

[49] Widrow B., Stearns S., Adaptive Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1985.

[50] Wozencraft J.M., Jacobs I.M., Principles of Communication Engineering, John Wiley and Sons Inc., New York, 1965.

[51] Zeng G., A new adaptive IIR algorithm and the convergence factors for digital and analog adaptive filters Ph.D. dissertation, University of New Mexico, 1988.

BIOGRAPHICAL SKETCH

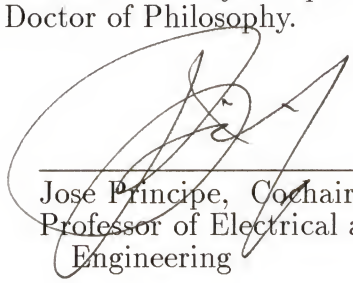
Jui-Kuo Juan was born on February 13, 1963 in Taichung, Taiwan. He completed his undergraduate study in 1986 at the National Central University, Taiwan. He received his master of science degree in Electrical Engineering from University of Florida in December 1993. Since then, he has worked with Dr. Harris. His research interests are analog VLSI circuit design and adaptive filter.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



John G. Harris, Chairman
Assistant Professor of Electrical and Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Jose Principe, Cochairman
Professor of Electrical and Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Robert M. Fox
Associate Professor of Electrical and Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



LiMin Fu
Associate Professor of Computer and Information Science and Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Mark Yang
Professor of Statistics

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

May 1998

Winfred M. Phillips
Dean, College of Engineering

Karen A. Holbrook
Dean, Graduate School